Oracle® DIVArchive

Installation and Configuration Guide

Release 7.5

E79752-05

August 2017

ORACLE®

Oracle DIVArchive Installation and Configuration Guide, Release 7.5

E79752-05

Primary Author: Lou Bonaventura

Contents

4 Start Configuring DIVArchive

5 Configuring DIVArchive Robot Managers

6 Configuring DIVArchive Actors

7 Configuring DIVArchive Managers

8 Configuring DIVArchive Checksum Support

9 Installing and Configuring Transcoders

10 Integrating Movie2Me with DIVArchive

11 Frequently Asked Questions

A DIVArchive Options and Licensing

B Dynamic Configuration Changes

C   Sources and Destinations Guide

D ADIC SDLC Installation and Configuration

List of Tables

Preface

This book describes initial and general installation and configuration of the Oracle DIVArchive Suite system. The manual assumes a working knowledge of the Windows and Linux operating systems, and additional concepts such as networking, RAID, tape drives, and fibre channel technologies.

Audience

This document is intended for the Oracle Installation Team, System Administrators, and system users.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the Oracle DIVArchive documentation set for this release located at https://docs.oracle.com/en/storage/#csm.

For information on Oracle Storage Cloud visit the following links.

For information regarding metered and non-metered accounts:
http://docs.oracle.com/en/cloud/get-started/subscriptions-cloud/csgsg/

For up to date Cloud information:
http://docs.oracle.com/cloud/latest/

For further assistance:
http://docs.oracle.com/cloud/latest/storagecs_common/index.html

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

This chapter describes the Oracle DIVArchive 7.5 release, and includes the following information:

- Release Overview
- Port Utilization
- New and Enhanced Features and Functionality
    - Oracle DIVArchive
    - Oracle DIVAnet
    - Oracle DIVArchive DIVAprotect
        * Events
        * Metrics
        * New Predefined Metrics
- DIVArchive 7.5 Configuration File Changes
- DIVArchive Media Storage Formats
    - AXF (Archive eXchange Format)
    - Tape Groups
    - Disk Arrays
- Complex Objects
- Core Software Components
- Additional Software Components
- Oracle DIVArchive Utilities

Release Overview

The DIVArchive architecture enables integration of many different types of servers and technologies, for example Broadcast Video Servers, Storage Area Networks, and Enterprise Tape Libraries. The DIVArchive installation varies from site to site, therefore the exact configuration of your specific DIVArchive platform is not described in this book. Refer to your *DIVArchive Site Configuration* document for details on your specific DIVArchive system installation and configuration.

The *Site Configuration* is a document prepared by Oracle (or an affiliated system integrator) after installation at your site. It is a complete record of the configuration of

the DIVArchive sub-systems and third party interfaces, site details and contacts, user names and passwords, and remote access codes.

The *Site Configuration* (for your site) is referenced at various points in this document, such as when user names and passwords are required for access to the DIVArchive software utilities.

> **Note:** The File System Interface is not released with DIVArchive 7.x and is only supported by special request.

See Appendix A for DIVArchive options and licensing.

Port Utilization

The following table lists the standard ports used by the DIVArchive system. If you need assistance contact Oracle Support.

Table 1–1 DIVArchive Port Utilization

Service	Port Number	Description and Notes
FTP	21/tcp	Port depends on configuration
Telnet	23/tcp	Solaris hosts only
SSH	22/tcp	Solaris hosts only
HTTP	80/tcp	DIVAdirector
Oracle SQLNet	1521/tcp	Manager database access
RDP (Microsoft Terminal Services)	3389/tcp	Remote Desktop access
VNC	5800/tcp 5900/tcp	Used only when AMC is installed to access the Avid servers.
Oracle DIVArchive Robot Manager	8500/tcp	Robot Manager
Oracle DIVArchive Manager	9000/tcp	Manager
Oracle DIVArchive Actor	9900/tcp	Actor
Flip Factory	9000/tcp	Flip Factory
DIVArchive AMC	6101/tcp	Avid AMC
Oracle DIVAnet	9500/tcp	DIVAnet

New and Enhanced Features and Functionality

The following sections describe new and enhanced features and functionality included in this DIVArchive release.

Oracle DIVArchive

This following list describes general DIVArchive new and enhanced features and functionality included in this release:

- All core DIVArchive components now support for Oracle Linux 7 x86_64 and later.
  - If you require a Linux environment in a language other than English, create a user and identify the desired language in the user profile. Oracle Linux 7 x86_

64 and later has support for a variety of languages (other than English) and you select the language during Linux installation.

- Windows installations *must* use the English language, and Oracle only supports English-based Windows environments.

- See the *Oracle DIVArchive Supported Environments Guide* in the *Oracle DIVArchive 7.5 Core documentation* Library for information about certain limitations when running in the Linux environment.

■ Archive and restore operations of symbolic links are now supported in Linux.

- Shortcuts created using the Windows operating system are not represented as symbolic links because they are treated as files. Only symbolic links created on the UNIX platform are archived and represented as symbolic links in DIVArchive.

- The Java and C++ APIs file list returned from a `getFilesAndFolders` call includes symbolic links.

- The export and import operations `type` attribute can now contain the letter *S* to represent a symbolic link.

■ DIVArchive 7.5 supports configuring Oracle Archive Cloud accounts for use.

- Any disks added to a cloud array are considered cloud disks.

- See Configuring Oracle Archive Cloud for DIVArchive for detailed information on using Oracle Archive Cloud accounts with DIVArchive.

- Visit the following links for information on the Oracle Storage Cloud:

  * For information regarding metered and non-metered accounts:

    `http://docs.oracle.com/en/cloud/get-started/subscrip-tions-cloud/csgsg/`

  * For up to date cloud information:

    `http://docs.oracle.com/cloud/latest/`

  * For further assistance:

    `http://docs.oracle.com/cloud/latest/storagecs_common/index.html`

■ The DIVArchive Control GUI contains the following enhancements:

- A new column is available in the *Disks View*. The **Consumed Size** column represents the space (in kilobytes) consumed by content on disk. The new column is especially useful for cloud accounts with unlimited disk space because it provides visibility into the amount of content stored in the cloud.

  The cloud storage class associated with the array is also displayed in this view. Non-cloud disks have a storage class of **NONE**. Cloud disks have a storage class of **Standard** (immediately available for download from the cloud) or **Archive** (requires at least 4 hours to download from the cloud).

- To view the storage class associated with a particular cloud instance, the *Object Properties View* contains a new column called **Cloud Storage Class**.

■ Database password encryption has been enhanced for additional security. See the *Oracle DIVArchive Database User's Guide* in the *Oracle DIVArchive 7.5 Core documentation* library for detailed information.

- DIVArchive 7.5 in a Windows environment only supports DIVArchive Oracle database package `OracleDivaDB_3-0-0_12_1_0_2_0_SE2_OEL7_ Windows_ 64-bit.zip`.

- DIVArchive 7.5 in a Linux environment only supports DIVArchive Oracle database package `OracleDivaDB_3-0-0_12_1_0_2_0_SE2_OEL7_x86_64.sh`.

- The Control GUI password hash has been enhanced for additional security.

- LTO-7 drives are now supported.

- DIVArchive operations now include the total tape size.

  - For some specific tape drives (Oracle T10K and IBM LTO drives) the Actor now returns the true remaining size on the tape, and the last written position on tape to the Manager during a transfer of content to tape. The remaining size is given in number of bytes of uncompressed data.

  - The Manager uses the remaining size and last written position to obtain the true total and remaining size on the tape.

  - The Manager uses the true total and remaining size on tape in all tape based operations instead of relying on the size of the tape type.

  - Export and Import operations now include the total tape size.

- DIVArchive 7.5 supports a new type of ODA drive from Sony. The details of these drives are as follows:

  - Sony released a new generation of ODA drives; ODS-280F and ODS-280U. DIVArchive has only been tested with the Fibre Channel type. These drives are twice as fast as the Gen1 drives. The ODS-280U has not been qualified for use with DIVArchive 7.5.

  - A new type cartridge is available for this drive; ODC3300R. This is a WORM drive with a 3.3 TB capacity.

  - Gen2 drives can read content written on Gen1 media with Gen1 drives. DIVArchive does not support the *read only* media-drive compatibility. Oracle recommends isolating Gen1 media from Gen2 media in the configuration (there is no cross-generation compatibility) and there must be at least one Gen1 drive in a library containing Gen1 cartridges.

- DataExpedition release 1.17 is now supported.

- The JDBC Thin Driver enables replacing the Oracle SID setting with the Oracle Service Name.

- MPEG2 Transport Stream enhancements for Oracle DIVArchive Partial File Restore operations. This feature now supports HD MPEG video essences with AES3 audio tracks.

- Linux-based Actors only support Telestream Vantage for transcoding operations.

- Promedia Carbon (formerly Rhozet) is now supported. This is a new transcoder supported in DIVArchive 7.5. You can now select the transcoder type **tre** from the Configuration Utility. Both the *Name* and *GUID* are supported as options for *Presets* and *Profiles* format types.

- Archive and Restore of empty files and folders are now supported.

  - *Empty files and folders are only supported by AXF*. When *Legacy* format is in use, DIVArchive reports an error if an empty file or folder is discovered during the transfer.

- Empty files are displayed in the *Elements* list on the **Instances** tab in the *Object Properties* dialog box.

- Folders are *not* displayed in the *Elements* list because they are only metadata and have no physical presence on the tape.

- All files and folders are displayed on the **Components** tab of the *Object Properties* dialog box.

■ Long path names are now supported in both Windows and Linux environments.

- Absolute path names are supported on both Windows and Linux up to a maximum of 4000 characters.

- Relative path names are limited to 256 characters on Windows systems (only).

■ The Control GUI displays the fully qualified path to a file, and the fully qualified path to an empty folder, in the *Components* list on the **Properties** tab of the *Object Properties* dialog box.

■ UNC paths are supported for SMB Source/Destinations and Managed disks if the UNC path is mounted directly on the Windows Actors.

■ Linux-based Actors support Telestream Vantage for transcoding operations.

■ Linux-based Actors cannot use CIFS, and a new path is required to enable Linux-based Actors to transfer content to and from SMB network shares (transcoder cache).

> **Caution:**   The sequence of the `actor`, `actorPath`, and `transcoder` settings is important. The parameter sequence must be `actor`, then `actorPath`, and then `transcoder`.

The path is specified as the value of the `actorPath` parameter in a delimited list as follows:

```
[actor:actor_name,actorPath:actor_path,transcoder:trancoder_ip_
address],cifs://username:password@\\transcoder_cache_ip_address\transcoder_
cache
```

You do not need to change an existing CIFS path for use with a pre-existing Vantage transcoder configuration if you are switching from a Windows Actor to a Linux Actor. The CIFS path is only used by Vantage to access the cache. The only thing that changes is adding the `actorPath` to the working directory of the transcoder. For example:

```
[actor:actor1,actorPath:/tmp/vantagecache,transcoder:100.100.100.100],cifs://A
dministrator:password@\\100.100.100.100\VantageStore
```

The `actorPath` represents the path the Linux Actor uses to transfer content to and from the SMB network share (transcoder cache) on a Linux system.

■ Migration service jobs now have events associated with them. All job events are displayed under the **Job Events** tab in *Job Properties* dialog box. By default, events are loaded in descending order by time and event id. The *Events* table in the **Job Events** tab highlights events with different colors based on severity. Red indicates an Error, yellow indicates a Warning, and white indicates Information. You click **Refresh** to refresh the entire *Job Properties* dialog box display.

You *must* use a migration job to change a tape format from Legacy to AXF. *Repacking a tape will not change the tape format*. Repacking of existing Legacy format

objects retains the format of the tape even if the tape group format was updated in the configuration from Legacy to AXF.

■ You can now change the status of SPM Failed Actions to Completed by right clicking the action and selecting Mark Action Completed from the context menu.

Normally, SPM will retry a completed Copy action if the *Once Only* option is set to **NO**, and a user manually (or accidentally) deletes the instance that SPM copied before the Storage Slot expires. Also, SPM will normally retry a completed Delete action if a user manually (or accidentally) copies an instance to the Storage Slot medium after SPM deleted it. *Actions marked as complete by a user will never be retried by SPM*. However, you can reschedule a user completed action by right clicking it and selecting **Reschedule Action** from the context menu. The **Mark Action Completed** (by a user) option is only available using the administrator profile. *See Appendix A for Oracle DIVArchive options and licensing information*.

Oracle DIVAnet

DIVAnet version 2.x is available for 64-bit Windows platforms, and 64-bit Oracle Linux. The DIVAnet 2.2 release is compatible with DIVArchive releases 7.3.1 and later. Currently, DIVAnet 2.2 requires that all sites in the DIVAnet network run DIVAnet release 2.2.

New and enhanced features and functionality in DIVAnet 2.x (and later) includes the following:

■ New architecture supports more sites, requests, connections and objects.

■ More configurable, manageable, flexible, and improved error handling.

■ New DIVAnet UI (User Interface) has more powerful request monitoring, asset search capabilities, and a simple, more flexible reconciliation, copy, and delete process.

■ API functionality has been enhanced.

■ Support for Oracle Linux 7 x86_64 and later.

Refer to the *Oracle DIVAnet Installation, Configuration, and Operations Guide* in the *Oracle DIVAnet documentation* library for more information on the new DIVAnet releases.

Oracle DIVArchive DIVAprotect

This section describes events and metrics processed by DIVAprotect.

Events

DIVAprotect Archive, Copy, Copy As, Restore, and Oracle DIVArchive Partial File Restore request events will populate the *Transfer Size* and *Duration* fields.

The ANALYZE_END and ANALYZE_ERR events are removed from the predefined DIVAprotect events.

Metrics

To standardize the *Metric ID* across all DIVArchive installations, all predefined metrics have a hard-coded *Metric ID* instead of using a database sequence. All user defined metrics start with *Metric ID* 1001. The database upgrade scripts handle this migration for you during upgrades from DIVArchive 7.4 to release 7.5. If you already had any user-defined metrics, they are automatically assigned a new *Metric ID* starting with 1001.

The following pre-defined metrics have been remove. However, there is no impact for earlier releases and customers currently using them; those metrics will continue to exist, update, and are not removed during the upgrade process to the 7.5 release. This only affects new installations of DIVArchive 7.5 and later.

- `ANALYZE_NUMBER_DAY`

- `ANALYZE_NUMBER`

- `ANALYZE_ABORTED_NUMBER_DAY`

- `ANALYZE_ABORTED_NUMBER`

- `ANALYZE_DATA_DAY`

- `ANALYZE_DATA`

- `ANALYZE_AVG_THROUGHPUT_DAY`

- `ANALYZE_AVG_THROUGHPUT`

- `ANALYZE_MIN_THROUGHPUT_DAY`

- `ANALYZE_MIN_THROUGHPUT`

- `ANALYZE_MAX_THROUGHPUT_DAY`

- `ANALYZE_MAX_THROUGHPUT`

- `ANALYZE_TIME_DAY`

- `ANALYZE_TIME`

- `MEDIA_OBJECT_INSTANCE_CREATE_DELETE_DAY`

- `MEDIA_OBJECT_INSTANCE_CREATE_DELETE`

New Predefined Metrics

The following new predefined metrics have been added in this release. See Metrics Definitions for a complete list.

- `ACTOR_READ_WRITE_MONTH`

- `ACTOR_READ_WRITE_NUMBER_MONTH`

- `ACTOR_TIME_ALL_OPERATION_MONTH`

- `ACTOR_TIME_READ_MONTH`

- `ACTOR_TIME_WRITE_MONTH`

- `DISK_AVG_TRANSFER_RATE_READ_MONTH`

- `DISK_AVG_TRANSFER_RATE_WRITE_MONTH`

- `DISK_CHECKSUM_FAILURE_COUNT_MONTH`

- `DISK_NUMBER_READ_ABORTED_MONTH`

- `DISK_NUMBER_READ_MONTH`

- `DISK_NUMBER_WRITE_ABORTED_MONTH`

- `DISK_NUMBER_WRITE_MONTH`

- `DISK_READ_MONTH`

- `DISK_TIME_ALL_OPERATION_MONTH`

- `DISK_TIME_READ_MONTH`

- DISK_TIME_WRITE_MONTH
- DISK_WRITE_MONTH
- DIVARCHIVE_SYSTEM_ACTIVE_ARCHIVE_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_ACTIVE_COPY_AS_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_ACTIVE_COPY_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_ACTIVE_RESTORE_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_AVG_READ_WRITE_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_NUMBER_OBJECT_ARCHIVE_MONTH
- DIVARCHIVE_SYSTEM_NUMBER_OBJECT_CREATED_MONTH
- DIVARCHIVE_SYSTEM_NUMBER_OBJECT_DELETED_MONTH
- DIVARCHIVE_SYSTEM_NUMBER_OBJECT_INSTANCE_COPY_MONTH
- DIVARCHIVE_SYSTEM_NUMBER_OBJECT_INSTANCE_CREATED_MONTH
- DIVARCHIVE_SYSTEM_NUMBER_OBJECT_INSTANCE_DELETED_MONTH
- DIVARCHIVE_SYSTEM_NUMBER_OBJECT_RESTORE_MONTH
- DIVARCHIVE_SYSTEM_OBJECT_EXPORT_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_OBJECT_IMPORT_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_OBJECT_INSTANCE_EXPORT_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_OBJECT_INSTANCE_IMPORT_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_READ_WRITE_ABORTED_NUMBER_MONTH
- DIVARCHIVE_SYSTEM_READ_WRITE_MONTH
- DIVARCHIVE_SYSTEM_READ_WRITE_NUMBER_MONTH
- MEDIA_ARCHIVED_OBJECT_DATASIZE_MONTH
- MEDIA_DATA_SIZE_DAY
- MEDIA_DATA_SIZE_MONTH
- MEDIA_OBJECT_INSTANCE_EXTERN_MONTH
- MEDIA_OBJECT_INSTANCE_ONLINE_MONTH
- MEDIA_READ_WRITE_MONTH
- MEDIA_READ_WRITE_NUMBER_MONTH
- MEDIA_RESTORE_OBJECT_DATASIZE_MONTH
- MEDIA_TAPE_EXPORT_NUMBER_MONTH
- MEDIA_TAPE_IMPORT_NUMBER_MONTH
- SD_ARCHIVE_OBJECT_DATASIZE_MONTH
- SD_READ_MONTH
- SD_READ_NUMBER_MONTH
- SD_RESTORE_OBJECT_DATASIZE_MONTH
- SD_TIME_MONTH
- SD_WRITE_MONTH

- SD_WRITE_NUMBER_MONTH

- TAPE_DRIVE_DATA_RATE_MONTH

- TAPE_DRIVE_ERROR_RATE_MONTH

- TAPE_DRIVE_NUMBER_READ_WRITE_ABORTED_MONTH

- TAPE_DRIVE_OPERATION_TOTAL_TIME_MONTH

- TAPE_DRIVE_READ_WRITE_MONTH

- TAPE_DRIVE_READ_WRITER_NUMBER_MONTH

- TAPE_DRIVE_TIME_ALL_OPERATION_MONTH

- TAPE_DRIVE_TIME_READ_MONTH

- TAPE_DRIVE_TIME_WRITE_MONTH

- TAPE_LIBRARY_NUMBER_DISMOUNT_ABORTED_MONTH

- TAPE_LIBRARY_NUMBER_MOUNT_ABORTED_MONTH

- TAPE_LIBRARY_NUMBER_MOUNT_MONTH

- TAPE_LIBRARY_NUMBER_READ_MONTH

- TAPE_LIBRARY_NUMBER_WRITE_MONTH

- TAPE_LIBRARY_READ_MONTH

- TAPE_LIBRARY_WRITE_MONTH

- TRANSCODE_DATA_MONTH

- TRANSCODE_NUMBER_MONTH

- TRANSCODE_TIME_MONTH

- MEDIA_OBJECT_INSTANCE_CREATE_DAY

- MEDIA_OBJECT_INSTANCE_DELETE_DAY

- MEDIA_OBJECT_INSTANCE_CREATE

- MEDIA_OBJECT_INSTANCE_DELETE

- MEDIA_OBJECT_INSTANCE_CREATE_MONTH

- MEDIA_OBJECT_INSTANCE_DELETE_MONTH

- MEDIA_OBJECT_INSTANCE_EXTERN_DATASIZE_DAY

- MEDIA_OBJECT_INSTANCE_EXTERN_DATASIZE_MONTH

- MEDIA_OBJECT_INSTANCE_ONLINE_DATASIZE_DAY

DIVArchive 7.5 Configuration File Changes

Changes have been made to the configuration files included with DIVArchive for release 7.5. The changes coincide with the new and enhanced functionality. The following is a brief overview of the changes; you can find more details in the sections specific to each component.

The new Manager ABORT_ARCHIVES_ON_EMPTY_FILES configuration setting terminates Archive requests that contain empty files or folders. The default setting is disabled; allowing the archiving of empty files or folders.

The new Manager `DIVAMANAGER_DBSERVICENAME` configuration setting is the Oracle `ServiceName` setting. This is set during database installation. The recommended setting is `lib5.world`. This value or the `DIVAMANAGER_DBSID` must be set. If both are set, the Service Name takes precedence over the Oracle SID.

DIVArchive Media Storage Formats

This section describe the media storage formats available in this DIVArchive release.

AXF (Archive eXchange Format)

AXF (Archive eXchange Format) is an open format that supports interoperability among disparate content storage systems and ensures the content's long-term availability no matter how storage or file system technology evolves.

An AXF object is an IT-centric file container that can encapsulate any number, and any type, of files in a fully self-contained and self-describing package. The encapsulated package contains its own internal file system, which shields data from the underlying operating system and storage technology. It's like a file system within a file that can store any type of data on any type of storage media.

Tape groups or disk arrays used by complex object requests must be in an AXF format, because complex objects cannot be stored in Legacy format. Because all complex objects are written in the AXF format, any instance of a complex object will also be in the AXF format.

Tape Groups

In DIVArchive, a *Tape Group* or *Disk Array* has a media format parameter that indicates which storage media format to use when creating Archived objects. You can set the media format to either *DIVArchive Legacy Format* or the *AXF Format*. This setting can be changed at any time and does not influence content already stored. It is possible to have more than one storage media format within tape groups and disk arrays.

A DIVArchive object instance is only written in one media format. Therefore, if an object spans tapes, each tape used as part of an object instance will be written in the same media format. An object can contain multiple instances, each of which can be stored in either Legacy or AXF format.

Although a tape group can contain more than one storage format, an individual tape has at most one storage media format. The format of a tape instance is the format of the tape on which the instance resides. *All instances on a tape must have the same format*.

The media format for an empty tape is assigned when the first object on that tape is written. The tape is assigned the format of the tape group that appears in the request. After the media format for a tape is assigned, you cannot change it unless all objects on the tape are deleted. After deletion of all objects from a tape, the tape's format becomes unassigned until content is again written to the tape. *If the tape was in use, the tape format cannot change unless it is empty and cleared.*

Both Legacy and AXF formatted tapes can exist in the same group. Nevertheless, objects in AXF format will only be written to AXF formatted tapes, and objects in Legacy format will only be written to Legacy formatted tapes, even though they are in the same tape group.

> **Note:** A Repack request will always write the destination tape in the same media format as the source tape.

Similarly, tape spanning operations will always use the same format across all tapes storing spanned objects. If an instance spans across multiple tapes, then all tapes used to span the content will have the same format.

Disk Arrays

Unlike tapes, disks do not have a format. DIVArchive allows storing objects in different media formats on the same disk. If a disk contains objects in Legacy format, and that disk is then assigned to an AXF formatted array, it will still contain objects in Legacy format. However, new objects written to the disk will be in AXF format.

If a disk instance is non-complex and permanent (not a cache instance), it is stored in the format of the destination array. If a cache instance is non-complex, it is stored in the format of the group specified in the request.

You can use the Copy To Group, or Copy As New requests to migrate objects from Legacy media format to AXF media format (or back). However some AXF objects cannot be copied to the Legacy format; copying objects from Legacy format to AXF format does not present any issues. In DIVArchive the only limitation on copying an object instance from AXF format to Legacy format is the complex object feature.

Complex Objects

Complex objects have significantly expanded the object component boundaries, allowing up to one million files and ten thousand folders per object.

Complex objects maintain information about files and folders in the archive. They store subtotals for each folder, including the total number of files and subfolders within the folder, and the total size of all files within the folder and within any subfolders.

DIVArchive uses the configurable *Complex Object Threshold* parameter during archival to determine whether a new object should be complex based on the number of components. This value is set in the `manager.conf` configuration file. If the number of components is greater than the *Complex Object Threshold*, the object becomes a complex object. After an object is identified as a complex object it will always be complex; even if it is copied using the Copy As command, or imported using the Export/Import Utility.

Oracle recommends that the threshold remain at the default value (1,000 components) unless there is a specific reason to adjust the value. Contact Oracle Support for assistance as required.

A complex object differs from a non-complex object in several key ways. For example, the file and folder metadata information of a complex object is stored in a file, not in the Oracle Database. The file contains the file names, folder names, checksums, and files sizes. The files are located in the Metadata Database root directory. *Complex objects must be stored in AXF format whether on tape or on disk*.

Complex objects can contain hundreds of thousands of files. However, some DIVArchive API commands (for example, `GetObjectInfo`) will not return the entire set of files. Instead, these commands return a single placeholder file which prevents downstream applications from being overwhelmed by file and folder information. Also, the entire set of files on a tape are not displayed in the Control GUI *Object Properties* and *Tapes* screens, only a single placeholder file is shown. The DIVArchive API includes a command to return all of the files and folders within a complex object. See the appropriate Oracle DIVArchive API documentation in the *Oracle DIVArchive documentation* libraries for details.

Oracle DIVAnet does not currently support replication of complex objects.

The following features do not support complex objects:

- *Delete on Source* option
- *Verify on Restore* (VFR) checksum feature
- *Verify on Archive* (VFA) checksum feature
- `deleteFile` API call
- `getObjectListbyFileName` API call
- `GetByFilename` API call (for Avid connectivity)
- `DeleteByFilename` API call (for Avid connectivity)

Complex Objects and FTP

When archiving complex objects using the FTP protocol, and an FTP Client with default settings (FileZilla is recommended), the transfer will typically fail when archiving any object with more than approximately 3,900 files.

Occasionally, during the directory scan, the Actor connection times out before the size of the object can be computed. More often, a request terminates in the middle of the transfer because the FTP server is consuming all of the available sockets.

You can add the following parameters in the *Source/Destination Command Options* or in the *Options* of the command itself to resolve timeout issues:

```
-transfer_timeout 1200
-list_timeout 600
```

See Appendix C for detailed parameter information.

Use the following procedure to include the parameters in the Source/Destination frame in the Configuration Utility:

1. Open the DIVArchive Configuration Utility.

2. Navigate to the **System** tab.

3. Double-click the desired Source/Destination in the *Sources and Destinations* frame to open the edit dialog box.

4. Add the two parameters (`-transfer_timeout 1200` and `-list_timeout 600`) in the *Connect Options* field.

5. Click **OK** to save the changes.

6. Notify the Manager of the changes using the `Control+N` key combination.

Oracle recommends setting the following corresponding parameters in the FileZilla server under **General Settings**:

Connections Timeout = `600`

No Transfer Timeout = `1200`

1. Open the FileZilla server interface.

2. Click the **Server Options** icon on the tool bar.

3. Adjust the settings in the **General Settings** area.

If requests terminate unexpectedly during transfers, adjust the Windows Registry parameters as follows:

1. Open `regedit`.

2. Modify (or create) the following values under `HKEY_LOCAL_ MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`:

   ```
   TcpTimedWaitDelay = 10
   MaxUserPort = 90000
   ```

   1. If the desired registry parameter does not exist, right-click and create a new double word (`DWORD`) value.

   2. If the parameter does exist, double-click it and enter the values.

3. Restart the computer to enable the new registry values.

Core Software Components

DIVArchive includes the core software components discussed in the following subsections.

Long path names are now supported on both Windows and Linux. Absolute path names are supported on both Windows and Linux to a maximum of 4000 characters. Relative path names are limited to 256 characters on Windows systems (only).

All Windows batch files (`.bat`) have corresponding shell scripts (`.sh`) in Linux. You must substitute Windows paths with Linux paths when operating on Linux. For example, the Windows path `C:\DIVA\Program` equates to `/home/diva/DIVA/Program` in the Linux environment.

> **Note:** Linux commands, paths, and file names are case-sensitive.

The following features require Windows-based Actors:

- Oracle DIVArchive Avid Connectivity

- Transcoder integration

- Tape Reading Utility

- You must use standard commands. For example `DD` and `MT`.

Linux Actors support QuickTime, GXF, and TeleStream wrappers for Oracle DIVArchive Partial File Restore (video). *See Appendix A for Oracle DIVArchive options and licensing information*.

Due to degraded performance, Windows IIS and FileZilla FTP sources and destinations cannot be used for complex objects. Oracle only supports Linux-based FTP servers when operating in a Linux environment. The Windows IIS and FTP servers cannot accommodate large numbers of files.

Oracle DIVArchive Manager

The DIVArchive Manager is the main component in a DIVArchive system. All archive operations are controlled and handled by the DIVArchive Manager. Operation requests are sent by initiator applications through the DIVArchive Client API. DIVArchive supports Main and Backup DIVArchive Managers (*see Appendix A for Oracle DIVArchive options and licensing information*).

Oracle DIVArchive Actor

The DIVArchive Actor is the data mover between devices in your production system. Actor supports interfacing and data transfer between many different types of devices.

All Actor operations are initiated and coordinated by the DIVArchive Manager through the TCP/IP networking protocol. Key benefits of the distributed design of the DIVArchive Actors are:

- You can expand the archive subsystem to increase the overall bandwidth by adding more Actors to the system. *See Appendix A for Oracle DIVArchive options and licensing information*.

- You can share SAN based disk and tape drive resources among multiple Actors.

- In combination with the DIVArchive Manager, multiple Actors provide scalability, load balancing, redundancy, and failover. You can take individual Actors offline for maintenance without shutting down the DIVArchive system.

> **Note:** UNC paths are supported for SMB Source/Destinations and managed disks if the UNC path is mounted directly on Windows Actors.

Oracle DIVArchive Client APIs

The DIVArchive Client APIs are a set of functions enabling external applications, acting as clients, to use the services offered by the DIVArchive system.

A library of client functions is provided with the selected API and must be linked to each DIVArchive client application. These functions encapsulate client commands into DIVArchive request messages sent over a TCP/IP connection to the DIVArchive Manager.

Currently available APIs are C++, Java, and Web Services (Oracle DIVA Enterprise Connect). Refer to the appropriate *Oracle DIVArchive API documentation* in the *Oracle DIVArchive Additional Features documentation* library, and the *Oracle DIVA Enterprise Connect documentation* library for more information.

Oracle DIVArchive Database

The DIVArchive software is bundled with an Oracle database installation. The database stores all information relating to the DIVArchive system including its configuration. SQL queries used by the Manager are optimized to support configurations with up to 58 million components.

When installing DIVArchive in a 64-bit environment, the latest 64-bit DIVArchive Oracle release must be installed to use 64-bit support.

DIVArchive 7.5 in a Windows environment only supports DIVArchive Oracle database package `OracleDivaDB_3-0-0_12_1_0_2_0_SE2_OEL7_ Windows_64-bit.zip` and later.

DIVArchive 7.5 in a Linux environment only supports DIVArchive Oracle database package `OracleDivaDB_3-0-0_12_1_0_2_0_SE2_OEL7_x86_64.sh`.

The Oracle database is not intended to be modified directly by customers, but rather by using Oracle utilities. *Direct modification of this database by customers through Oracle utilities is not supported by Oracle*.

Oracle DIVArchive Metadata Database

DIVArchive stores object metadata separately from the Oracle database in the *DIVArchive Metadata Database*. The metadata database contains files stored in a file system local to the DIVArchive Manager. The files are located in the Metadata Database root folder. This storage method enables DIVArchive to effectively operate with large volumes of files, folders and other metadata.

The metadata database is very high performance, and has almost unlimited scalability. You must treat the metadata database with the same caution as the Oracle database, and it must be backed up at regular intervals through the DIVArchive Backup Service.

Oracle DIVArchive Control GUI

The DIVArchive Control GUI connects to both the DIVArchive Manager and the DIVArchive database. You use it to monitor, control, and supervise operations in DIVArchive. you can operate multiple Control GUI instances simultaneously from any computer that has TCP/IP connectivity to both the DIVArchive Manager and the DIVArchive database.

The Control GUI is not intended for the intensive archive operations of a DIVArchive system. Archive operations are typically initiated to DIVArchive from a Broadcast Automation or MAM (Media Asset Management) system.

See the *Oracle DIVArchive Operations Guide* in the *Oracle DIVArchive Core documentation* library for more information on using the interface.

The refresh rate for the Control GUI is set in the **Manager Setting** tab of the Configuration Utility in the *GUI: Dashboard Refresh Delay* field.

Additional Software Components

Additional modules are available to expand your DIVArchive system capabilities. Most of these options are currently covered in separate documents, but are briefly described here for completeness. *See Appendix A for DIVArchive options and licensing information*.

Oracle DIVArchive Robot Manager

You can use DIVArchive to only manage disk storage, but storage capacity can be further expanded by adding one or more tape libraries. In these cases, the DIVArchive Robot Manager module provides an intermediate software layer for the DIVArchive Manager to interact with many different types of tape libraries. It is connected to the DIVArchive Manager through TCP/IP. *See Appendix A for DIVArchive options and licensing information*.

This distributed architecture provides substantial flexibility including:

■ Libraries controlled using a SCSI interface are limited by the cable length. Because the connection to the DIVArchive Robot Manager from the DIVArchive Manager is over TCP/IP, the library does not need to be colocated near the DIVArchive Manager host computer.

■ Enabling installation of multiple, or dissimilar, libraries by configuring additional DIVArchive Robot Manager modules.

■ Enabling rapid development to support new types or models of libraries.

■ You can restart the robotics interface without needing to restart the DIVArchive Manager.

The DIVArchive Robot Manager interfaces with the library using either a direct interface to the library itself (through native SCSI, or SCSI over Fibre Channel), or through an intermediate Ethernet connection to the manufacturer's own library control software.

Oracle DIVArchive VACP (Video Archive Communications Protocol) Service

VACP (Video Archive Communications Protocol) is developed by Harris Automation Solutions and used by some automation systems for interfacing to an archive system. DIVArchive has its own API for communicating with the DIVArchive Manager, which is not compatible with VACP.

To provide interoperability without the need to redevelop the archive interface at the automation level, this module is provided to act as an interface to convert VACP commands from the attached automation system to DIVArchive API commands on computers that have TCP/IP connectivity to DIVArchive.

Oracle DIVArchive SPM (Storage Plan Manager)

The SPM (Oracle DIVArchive Storage Plan Manager) provides automatic migration and life cycling of material within the archive, based on the rules and policies defined in the SPM configuration. The DIVArchive DSM (Disk Space Monitor) works with SPM to delete material from SPM managed arrays (based on disk space watermarks).

Oracle DIVArchive SNMP (Simple Network Management Protocol) Agent

The DIVArchive SNMP (Simple Network Management Protocol) interface supports status and activity monitoring of different DIVArchive components. DIVArchive MIB (Management Information Base) is provided to third party SNMP monitoring applications. *The SNMP Agent uses the Windows SNMP Service and has not been ported to the Linux environment*.

Oracle DIVArchive Access Gateway and Oracle DIVAnet

The DIVArchive Access Gateway provides DIVArchive client authentication and authorization. It can act as an intermediate gateway between DIVArchive components (for example the VACP converter) or third party applications and the DIVArchive Manager, and can restrict that component or application from access to the DIVArchive system.

Oracle DIVAnet is a powerful feature that allows multiple DIVArchive platforms to exchange archive resources and content, whether the archive systems are local to each other or remote.

The Access Gateway is used in DIVAnet installations and is the portal for multiple DIVArchive systems to communicate with each other. See the *Oracle DIVAnet Installation, Configuration, and Operations Guide* in the *Oracle DIVAnet 2.2 documentation* library for more information.

Oracle DIVArchive DFM (Drop Folder Monitor)

The DIVArchive DFM (Drop Folder Monitor) provides automatic monitoring of newly created files in multiple local directories or FTP folders (or combinations thereof). One file, or multiple files, per DIVArchive object are supported. When a new file is identified, DFM issues an archive request automatically to DIVArchive to archive the new file. After the files are successfully archived, they are then automatically deleted from the source. Refer to the *Oracle DIVArchive Drop Folder Monitor (DFM) User's Guide*

in the *Oracle DIVArchive 7.5 Additional Features documentation* library for more information.

When DFM is used in a Linux environment to monitor an FTP folder, you must configure it as in the following example:

User
```
diva
```

User Home Directory
```
/ifs
```

Folder to be monitored
```
/ifs/folder1
```

A correct DFM configuration with these parameters is:

```
ftp://diva:password@host_ip/folder1
```

An incorrect DFM configuration with these parameters is:

```
ftp://diva:password@host_ip/ifs/folder1
```

Oracle DIVArchive Transcoder Support

The DIVArchive Actor can integrate with a transcoder engine to provide real time transcoding of material as it is archived or restored, or to create objects from already existing content within the archive. Currently, integration to Bitscream products, and Telestream Flip Factory, and Telestream Vantage are supported. However, multiple transcoders are only supported for Vantage.

> **Note:** DIVArchive 7.5 ends Oracle support for Telestream Flip Factory. Oracle will provide best efforts to assist customers to transition to other transcoding solutions.

Linux-based Actors only support Telestream Vantage for transcoding operations.

DIVArchive assumes a local transcoder address of `127.0.0.1` if a transcoder address is not specified in the transcoder's working directory.

The Promedia Carbon (formerly Rhozet) transcoder is supported in DIVArchive. You select the transcoder type **tre** from the Configuration Utility to use this transcoder. Both the *Name* and *GUID* are supported as options for *Presets* and *Profiles* format types.

Oracle DIVArchive Avid Connectivity

The following sections describe general Avid connectivity with DIVArchive.

See the *Oracle DIVArchive Avid Connectivity User's Guide* in the *Oracle DIVArchive Additional Features documentation* library for more information. *Also see Appendix A for DIVArchive options and licensing information.*

Avid DHM (Data Handler Module) Interface

The Avid DHM (Data Handler Module) interface support in DIVArchive enables finished content to be shared between post-production Avid environments and On Air Video servers. This eliminates the need for tape based content exchange. Timecode

based Partial File Restores of content to On Air environments, and finished Avid Sequence submissions to On Air servers are key to the DHM functionality offered within DIVArchive. DHM support is implemented in DIVArchive TMC (Transfer Manager Communicator).

Avid DET (Dynamically Extensible Transfer) Interface

The Avid DET (Dynamically Extensible Transfer) interface support in DIVArchive allows storage expansion of Avid Unity infrastructures and enables editors to move native Avid content in and out of the DIVArchive storage system. Partially edited content stored within DIVArchive through the Avid DET interface can be later restored to Unity, and an editor can then resume editing at the point where they stopped. DIVArchive stores these files in native Avid format. DET support is implemented in DIVArchive TMC (Transfer Manager Communicator).

Archive Manager Interface

An interaction between the Avid Archive Manager solution and DIVArchive is implemented in a separate service called AMC (Archive Manager Communicator). AMC handles Archive, Restore, Partial File Restore, and Delete commands from the Avid Archive Manager using DIVArchive to store Avid content in its native MXF OP1 Atom format.

Oracle DIVArchive DIVAprotect

The DIVAprotect option is a utility that collects operational statistics from the DIVArchive system to monitor and maintain the archive's subcomponents (servers, media, drives, tapes, and so on). Analysis of these statistics allows both proactive and reactive maintenance of the DIVArchive system. See the *Oracle DIVArchive DIVAprotect User's Guide* in the *Oracle DIVArchive Additional Features documentation* library for more information.

Oracle DIVArchive OTU (Object Transfer Utility)

The OTU (Object Transfer Utility) is an optional feature of the Control GUI providing a drag and drop interface to archive and restore content between DIVArchive and a (supported) source or destination server. *See Appendix A for DIVArchive options and licensing information*.

Oracle DIVArchive Utilities

The following sections describe utilities available in the DIVArchive system

Oracle DIVArchive Configuration Utility

You use the DIVArchive Configuration Utility to configure the DIVArchive system. It connects directly to the DIVArchive database, and can be run on any computer with TCP/IP connectivity to the host running the DIVArchive database.

Oracle DIVArchive Robot Manager Utilities

During configuration and troubleshooting of the library and its tape drives, DIVArchive provides both a command-line interface and GUI utility to send commands directly to the tape library through the Robot Manager. These utilities are not (and must not be) used while the DIVArchive Manager is running because this can

adversely affect archive operations. *See Appendix A for DIVArchive options and licensing information*.

Oracle DIVArchive Backup Service

The DIVArchive Backup Service ensures reliability and monitoring of both the Oracle Database and Metadata Database backups.

The DIVArchive Backup Service component is installed as an integral part of the standard DIVArchive system installation. The component is typically installed on the same server as the DIVArchive Manager and Oracle Database. The DIVArchive Backup Service enables configuration of scheduled backups through its configuration file. The DIVArchive Backup Service manages and monitors the entire backup process.

See the *Oracle DIVArchive Database User's Guide* in the *Oracle DIVArchive Core documentation* library for more information.

Oracle DIVArchive Scandrive Utility

This utility is provided on both Windows and Linux platforms. It assists in obtaining detailed device information such as serial numbers, firmware releases, and SCSI information from tape libraries or tape drives for use in the DIVArchive configuration.

Oracle DIVArchive Tape Reading Utility

> **Caution:** This utility *must not* be used while the DIVArchive Manager is running.

This utility is provided on both Windows and Linux platforms and is primarily used with the Robot Manager Client utilities to send manual Eject commands to a tape drive connected to an Actor. This utility also provides advanced tape based operations, such as tape formatting, but should only be used under guidance from Oracle Support.

The Tape Reading Utility is only supported by Windows-based Actors. You must use standard commands, for example, `DD` and `MT` when operating in a Linux environment.

Oracle DIVArchive DIVAscript

This utility allows DIVArchive C++ API commands to be executed using UNIX or DOS based scripts. It is designed to run automated tasks for testing rather than for any intensive uses. There is no Linux-based DIVAscript release.

Oracle DIVArchive RDTU (Recover Damaged Tape Utility)

The DIVArchive RDTU (Recover Damaged Tape Utility) is designed to recover object instances contained on a damaged tape. The utility can recover instances that have valid copies on other available media (that is, internal tape, or a connected disk or array) within a local or remote DIVArchive system.

2

Installing DIVArchive

This chapter describes DIVArchive software components and system installation, and includes the following information:

- Software Component Relationships
- DIVArchive Backup Service
- Installing the DIVArchive System
    - Downloading the Software
    - Installing DIVArchive in a Windows Environment
    - Installing DIVArchive in a Linux Environment
    - Starting, Stopping, and Accessing DIVArchive in Linux

Software Component Relationships

The following figure displays the relationships and dependencies among the software components of a DIVArchive system. It specifically points out the client/server links between them.

A client/server link between two components does not necessarily mean that the server software must be started before the client. For example, the DIVArchive Manager to Actor connection. Each Actor acts as a server and the Manager initiates a client connection to the Actor. However, an Actor can be launched after the Manager is running since the Manager will attempt to reconnect to the Actor at periodic intervals.

See Appendix A for DIVArchive options and licensing information.

> **Note:** DIVArchive can run independently of the Control GUI and Configuration Utility. They can be launched at any time after the DIVArchive Manager is running.
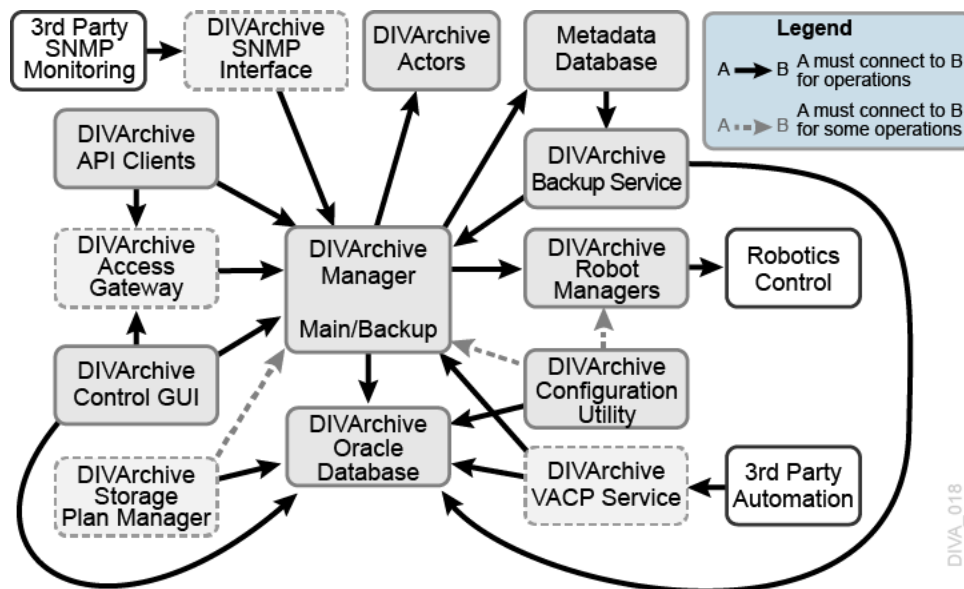
Software Component Distribution

The DIVArchive platform is flexible and scalable, so the installation of some software components can vary depending on the degree of storage and servers that are managed. Small installations can have all DIVArchive software components installed on a single computer. A very large installation will have these components distributed among several servers. All of these components run as system services.

The following list identifies where the components are typically installed:

DIVArchive Managers
Main and Backup DIVArchive Manager servers

DIVArchive Oracle Database
Main and Backup DIVArchive Manager servers

DIVArchive Metadata Database
Main and Backup DIVArchive Manager servers

DIVArchive Backup Service
Main and Backup DIVArchive Manager and Actor servers

DIVArchive Robot Managers
Main and Backup DIVArchive Manager servers. Robot Managers can also be installed on a separate server when the tape library is installed a substantial distance from the DIVArchive Manager servers.

DIVArchive Storage Plan Manager
Main and Backup DIVArchive Manager servers

DIVArchive VACP Services
Main and Backup DIVArchive Manager servers

DIVArchive SNMP Agent
Main and Backup DIVArchive Manager servers

DIVArchive Access Gateway
Main and Backup DIVArchive Manager servers

DIVArchive Actors
DIVArchive Actor servers

DIVArchive Transfer Manager Communicator (TMC)
DIVArchive Actor servers

DIVArchive Archive Manager Communicator (AMC)
DIVArchive Actor servers

DIVArchive Drop Folder Monitor (DFM)
DIVArchive Actor servers

DIVArchive Backup Service

> **Caution:** Users should have an elevated awareness of error messages from the Backup Service. See the *Oracle DIVArchive Database User's Guide* in the *Oracle DIVArchive 7.5 Core documentation* library for more information.

The DIVArchive Backup Service ensures reliability and monitoring of both the Oracle Database and Metadata Database backups.

The DIVArchive Backup Service component is installed as an integral part of the standard DIVArchive system installation. The component is typically installed on the same server as the DIVArchive Manager and Oracle Database. The DIVArchive Backup Service enables configuration of scheduled backups through its configuration file. The DIVArchive Backup Service manages and monitors the entire backup process.

When using complex objects, it is *strictly required* to use the Backup Service. The DIVArchive Backup Service is the only component backing up the Metadata Database and removing outdated Metadata files. When a delete request for a complex object is sent and processed, the data is removed from the Oracle Database, but the Metadata Database file is not deleted. The Metadata Database file is removed by the Backup Service after the configured clean up period (define by the ***Recovery Period*** parameter) has been reached.

> **Caution:** Do not change the *Metadata Location* parameter when the system is running.

If a database or system failure occurs, where restoring from a system backup is necessary, restoration of a stored backup is accomplished manually through existing Oracle scripts and should be performed by Oracle Support personnel only.

The DIVArchive Backup Service uses existing Oracle RMAN backup scripts to generate full database backups and incremental database backups. Oracle Database backups and Metadata Database backups are incrementally replicated to all remote backup systems by the DIVArchive Backup Service.

The DIVArchive Backup Service periodically sends status messages to the DIVArchive Manager. The DIVArchive Manager saves all error messages received in the Manager Events Log, and also forwards messages to all connected Control GUI applications to be displayed in a dialog box. If no Control GUIs are connected at the time of the error, no error dialog boxes will be displayed, but errors can be reviewed later in the Events Log.

You can configure the service to monitor specific disks for space and send warnings and errors accordingly. By default, the monitored disks are `C:` and `H:` drives. You can change this configuration modifying the `MONITORED_DRIVES=d1:,d2:` parameter in the Backup Service configuration file.

`MONITORED_DRIVES=d1:,d2:`
Identifies the drive letters to be monitored buy the DIVArchive Backup Service. The default is C: and H: and may be changed as required. The `d1` and `d2` represent the drive letters to backup.

Set the value for the Backup Service monitor timeout in the Configuration Utility **Manager Setting** tab. The default setting is fifteen minutes.

See the *Oracle DIVArchive Database User's Guide* in the *Oracle DIVArchive 7.5 Core documentation* library for more information.

Installing the DIVArchive System

The following sections describe installation of the DIVArchive system. Contact Oracle Support if you need assistance.

Downloading the Software

You must stay current with the release of DIVArchive that you install and operate. Current releases of the software are found on the Oracle Software Delivery Cloud.

Use the following procedure to obtain the DIVArchive software:

1. Log in to the Oracle Software Delivery Cloud (`https://edelivery.oracle.com`) and search for *DIVArchive*.

2. Select the licenses you require (for example, DIVArchive Actor, DIVArchive Manager, and so on). You must search each time after adding a new license to the list.

3. Select the operating system you run for each selected license using the **Select Platform** button.

4. Continue through the download wizard, accepting the terms, until the final download screen appears.

5. Confirm that all the licenses you require are listed.

6. Click **Download All** on the bottom right of the screen, or click the file name link, to download the software.

7. Save the download where it is easily accessible.

Installing DIVArchive in a Windows Environment

Use the following procedure to install DIVArchive on a Windows computer:

1. Locate the executable file on your computer.

2. Double-click the file to begin the installation.

3. After installation begins, select the components to be installed and then click Next to proceed.

4. Enter the desired installation folder in the *Destination Folder* field. Oracle highly recommends using the default installation folder (`C:\DIVA`). However, if another location is desired, click **Browse** to navigate the computer to locate the folder.

5. Click **Install** to proceed.

6. Installation will continue in the specified destination folder using the selected components. The installation progress screen is displayed until installation is complete. Clicking **Show Details** will show the detailed progress (per file) of the installation.

7. The **Close** button will be highlighted when the wizard is finished.

8. Click **Close** to complete installation and close the program.

Installing DIVArchive in a Linux Environment

Installing DIVArchive in a Linux environment is a manual installation and is somewhat different than the Windows installation procedure. The following sections describe installation procedures for DIVArchive 7.5 on a Linux host computer.

Prerequisites and Initial Set-up

These instructions assume that Oracle Linux 7, x86_64 or later, is installed with sqlplus, and the Oracle client.

If you require a Linux environment in a language other than English, create a user and identify the desired language in the user profile. Oracle Linux 7 x86_64 and later has support for a variety of languages (other than English) and the language can be selected during Linux installation.

Windows installations must use the English language and Oracle only supports English-based Windows environments.

For more information on Oracle Linux 7 x86_64 and later see the documentation located at https://docs.oracle.com/en/operating-systems/?tab=2, or contact Oracle Support for assistance.

> **Note:** Linux paths, file names are case-sensitive.

Use the following procedure to prepare for installation:

1. Use the following command to create a directory on the host computer:

```
mkdir /home/oracle/Downloads/DIVA_INSTALL
```

2. Copy the installation packages to the directory.

3. Confirm you have the latest DIVArchive and DIVArchive API (7.5 or later) releases and copy them into the directory you created in Step 1. The file transfer can take a bit of time due to the large file size.

If the shared memory on the server where the Oracle Database is installed is less than 16 GB, you must set it to at least 70 percent of your RAM.

1. Use the following command to confirm the computer's RAM size:

```
# free -m
```

The output will look similar to the following:

```
        total        used        free      shared  buff/cache   available
Mem:    15791         186       15456           8         148       15516
Swap:   16380           0       16380
```

2. Use the following command to check your shared memory setting in MB:

```
# df -m /dev/shm
```

The output will look similar to the following:

```
Filesystem      1M-blocks  Used Available Use% Mounted on
tmpfs              7896       0   7896     0%  /dev/shm
```

3. To change the size of shared memory you must add the following line into `/etc/fstab`. The setting must not exceed the size of your installed memory. You must restart the computer after making this change for it to take affect.

For example, the following command will increase the size of `/dev/shm` to 11GB:

```
tmpfs /dev/shm tmpfs defaults,size=11g 0 0
```

Creating the Oracle Database Partitions

First you must configure the drive partitions for the Oracle Database as follows:

1. Navigate to **Applications**, and then **Utilities**.

2. Click **Disks** from the menu.

3. Locate your disk in the Disks dialog box. Selecting the disk will display the *Device Name*.

4. In Linux you must add the disk (that you want to add partitions to) to the partition table using the `fdisk` utility. For example, `fdisk /dev/xvdb1`. You can use the `g` and `w` options to add it to the partitions table.

5. Click the **Plus** button on the right side of the Disks dialog box to add a partition.

6. When the Create Partition dialog box appears create the following four partitions. For each partition leave the *Erase* option and *Type* option at their default settings, and then click **Create**. Repeat this step for each partition.

   /u01
   This partition must be 10 GB in Linux. Use the operating system default block size.

   /u02
   This partition must be 30 GB in Linux. Oracle recommends using an 8 KB cluster size.

   /u03
   This partition must be 5 GB in Linux. Oracle recommends using a 4 KB cluster size.

   /u04
   This partition must be either 100 GB or all of the remaining disk space. Oracle recommends using a 64 KB cluster size.

7. When you are done creating the partitions and returned to the Disks dialog box, click the **Gears** icon on the right side of the screen.

8. Click **Edit Mount Options**.

9. Change *Automatic Mount Options* to **OFF**.

10. Select the *Mount at startup* check box.

11. Enter the appropriate mount point in the *Mount Point* field for that specific partition (`/u01`, `/u02`, `/u03`, `/u04`).

12. Click **OK**.

13. When this is completed successfully, all four partitions are identified and displaying their appropriate mount points in the Disks dialog box.

Use the following procedure for the *Managed Disk* partition (this must be 54 GB):

1. Locate the *Managed Disk* in the Disks dialog box.

2. Click the **Gears** icon on the right side of the screen.

3. Click **Format**.

4. Leave all settings at their defaults, but enter `/managed` in the *Mount Point* field.

5. Click **Format**.

6. When asked, click Format to confirm that you want to format the disk.

7. Click the **Gears** icon.

8. Click **Edit Mount Options**.

9. Change *Automatic Mount Options* to **OFF**.

10. Select the *Mount at startup* check box.

11. Enter `/managed` in the *Mount Point* field.

12. Confirm that the *Filesystem Type* is set to **ext4**.

13. Click **OK**.

Installing FTP Services

Use the following procedure to install FTP services on the host computer:

1. Open a terminal console.

2. At the prompt enter `yum install vsftpd.x86_64` and press **Enter**.

3. When prompted if it is ok to install, enter `y` and press **Enter**.

4. When installation is complete, start the service and confirm that it starts on system startup using the following commands:

```
service vsftpd start
chkconfig vsftpd on
```

5. Create a directory in the `/home/diva` path for managed storage and then mount the `/managed` partition in this location as follows:

```
mkdir /home/diva/managed
mount --bind /managed /home/diva/managed
```

Installing the Oracle DIVArchive Database

Confirm your RAM settings using the commands in Prerequisites and Initial Set-up. After you confirm the RAM settings, use the following procedure to install the Oracle DIVArchive database:

1. Open a terminal console.

2. If you run in a Virtual Machine (VM), confirm that your host name is in the `/etc/hosts` file using the following command:

```
gedit /etc/hosts
```

If the hosts file looks similar to this:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

You must replace `localhost` with your host name. For example, if the host name is `clefvm015L`, it will look like this:

```
127.0.0.1   celfvm015L localhost.localdomain localhost4
localhost4.localdomain4
::1         celfvm015L localhost.localdomain localhost6
localhost6.localdomain6
```

3. If you made changes to the host file save the changes and exit `gedit`.

4. Change to the directory of the shell script for the Oracle Database Package.

5. Change the permissions on the shell script using the following command to make it an executable file:

```
chmod +x OracleDivaDB_3-0-0_12_1_0_2_0_SE2_OEL7_x86_64.sh
```

6. Execute the script as follows:

```
./OracleDivaDB_3-0-0_12_1_0_2_0_SE2_OEL7_x86_64.sh
```

If an Oracle operating system account has already been created, you may be asked whether you want to change the password. Follow the prompts if you require a password change for this account.

7. When prompted for a `SYS` account password, ensure you use a secure password.

If at some point during the installation you receive the following error:

```
[FATAL] [INS-35172] Target database memory (5181MB) exceeds available shared
memory (3866MB) on the system
```

You must run the commands below to extend your `tmpfs` partition (if it is still not large enough):

1. Check the current size of the `tmpfs` partition:

```
df -h /dev/shm
```

2. Extend the amount of the target database memory size as follows:

   1. Execute `gedit /etc/fstab`.

   2. Add the following line to the bottom of the file:

   ```
   tmpfs /dev/shm tmpfs defaults,size=6G 0 0
   ```

   3. Save the file and exit `gedit`.

3. Execute the following commands:

```
umount tmpfs
mount -a
```

4. If the commands in Step 3 do not work, restart the computer and run the `df -h /dev/shm` command again to check that the size of `tmpfs` has actually increased.

5. Run the Oracle Database shell script again.

Installing DIVArchive 7.5

Use the following procedure to install DIVArchive 7.5 for Linux:

1. Open a terminal console.

2. Use the following command to change the permissions and make the installation script executable:

    ```
    chmod +x DIVArchive-7.5.{build_number}.sh
    ```

    The `{build_number}` in this command will be the last two digits of the file name. For example, in `DIVArchive-7.5.0.17.sh`, the `0.17` is the build number

3. Use the following command to execute the installation script:

    ```
    ./DIVArchive-7.5.{build_number}.sh
    ```

4. When the `Please specify diva user home directory [/home/diva]` prompt is displayed, press **Enter** to accept the default directory.

5. When the installation is complete you will be returned to the command prompt.

Creating the Database User

The database user must be created using the `DIVA` operating system user account. Use the following procedure to create the database user:

1. Open a terminal console.

2. Change to the `/home/diva/DIVA/Program/Database/Core/Install` directory.

3. Use the following command to change the permissions and make the installation script executable:

    ```
    chmod +x create_diva_user.sh
    ```

4. Execute the script using the following command, and replacing the `{IPADDRESS}` in the parameter with the IP address of your system:

    ```
    ./create_diva_user.sh {db_sys_password} {diva_dbuser} {diva_dbuser_password}
    {IPADDRESS}:1521/lib5.world
    ```

5. When the script completes the database user account will be active.

Installing the DIVArchive Services

You control the DIVArchive services using the `divaservice` script. The following are valid `divaservice` script parameters:

command
Can be one of the following: `configure`, `install`, `start-all`, `stop-all`, `start`, `stop`, `restart`, `uninstall`, `status`, or `list`. You can display the usage information of this list by running `./divaservice` with no additional parameters. See the following list for descriptions of each command.

SERVICE
Can be one of the following: `manager`, `actor`, `robotmanager`, `migrate`, `dfm`, `dbbackup`, `lynxlocaldelete`, `spm`, or `rsync`. You can display the usage information of this list by running `./divaservice` with no additional parameters.

SERVICE_NAME
The name of any currently installed DIVArchive service. You can view the services by running `./divaservice list`.

configuration file
This must be the absolute path to a valid configuration file.

The following are valid divaservice commands:

configure
This command expects a valid *SERVICE_NAME* and configures the specified (already installed) DIVArchive service.

install
This command expects a valid *SERVICE_NAME* and configuration file path. It installs the specified DIVArchive service using the specified configuration file.

start-all
This command starts all currently installed DIVArchive services.

stop-all
This command stops all currently installed DIVArchive services.

restart-all
This command restarts all currently installed DIVArchive services.

start
This command expects a valid *SERVICE_NAME* and starts the specified DIVArchive service.

stop
This command expects a valid *SERVICE_NAME* and stops the specified DIVArchive service.

restart
This command expects a valid *SERVICE_NAME* and restarts the specified DIVArchive service.

uninstall
This command expects a valid *SERVICE_NAME* and uninstalls the specified DIVArchive service.

status
This command expects a valid *SERVICE_NAME* and returns the status of the specified DIVArchive service.

list
This command lists the names of all currently installed DIVArchive services.

profile
This checks the DIVArchive services profile.

Use the following procedure to install and configure the DIVArchive services:

1. Open a terminal console.

2. Change to the /home/diva/DIVA/Program directory.

3. Execute the divaservice script using the following options for each of these conditions:

   ■   Available service names to use with the script are manager, actor, robotmanager, migrate, dfm, dbbackup, lynxlocaldelete, spm, and rsync.

- The script is located in the `/home/diva/DIVA/Program/` directory. Use the path in the command if you are not already in this directory.

  The first time you install a service you must use the `configure` option to include the configuration settings. It will generate a configuration file, install, and then start the service.

  ```
  divaservice configure {SERVICE_NAME}
  ```

- If you already have a fully configured (configuration) file, use the install option and include the absolute path to the configuration file for that service:

  ```
  divaservice install {SERVICE_NAME} {configuration_file_absolute_path}
  ```

- To start, stop, or restart all of the service at the same time use one of these options:

  ```
  divaservice {start-all|stop-all|restart-all}
  ```

- To start, stop, restart, uninstall, or get the current status of a specific service use one of these options:

  ```
  divaservice {start|stop|restart|uninstall|status} {SERVICE_NAME}
  ```

- Use the following command to display a list of services:

  ```
  divaservice list
  ```

- Use the following command to display the service profile:

  ```
  divaservice profile
  ```

- If you are upgrading, or want to install the services with preconfigured configuration files, you can use the `&&` command to do it consecutively (linking them together):

  ```
  divaservice install manager
  '/home/diva/DIVA/Program/conf/manager/manager.conf' && divaservice install
  actor '/home/diva/DIVA/Program/conf/actor/actor.conf' && divaservice
  install robotmanager '/home/diva/DIVA/Program/conf/robot_
  manager/robotmanager.conf'
  ```

- Use the following command to start the DIVArchive services when Linux starts:

  ```
  chkconfig {SERVICE_NAME} on
  ```

  For example, `chkconfig DIVAmanager_manager80` on will cause the Manager service to start with Linux.

For example, to install the Actor service use the following command. You can accept the default settings by pressing **Enter**.

```
divaservice configure actor
```

The displayed output will resemble the following:

```
Configuration File [actor.conf]:

# DIVA Actor connection port number
DIVAACTOR_PORT [9900]:

# DIVA Actor service name
# This is required when multiple actors are running on the same server by giving
```

```
different names for each actor.
# If this variable is set, the service name will be "DIVAactor_<SERVICE_NAME>" and
the name of actor will be "<SERVICE_NAME>".
# Default: If this variable is unset, the service name will be "DIVAactor_<host_
name>"
SERVICE_NAME [celfpdovm015L]: actor80

Adding tape group to diva user.  Setting will take affect after logout and login
Adding DIVAactor_actor80 tcp port 9900 to firewalld default zone
Enabled DIVAactor_actor80 as a service...
Started DIVAactor_actor80 as a service...
```

Creating Control GUI and Configuration Utility Shortcuts

You can add Control GUI and Configuration Utility shortcuts to your desktop (for easy access) using the following procedure:

1. Open a terminal console.

2. Open the gedit program with root user permissions. If you are not logged in as the root user, use the following command:

   ```
   sudo gedit
   ```

3. To create the Control GUI shortcut, enter the following text and save the file as /usr/share/applications/diva-control-gui.desktop:

   ```
   [Desktop Entry]
   Version=7.5
   Name=Control GUI
   Comment=Oracle DIVArchive CSM
   Exec=sh -c "cd /home/diva/DIVA/Program/GUI/bin/ && ./gui.sh"
   Icon=/home/diva/DIVA/Program/GUI/bin/gui.ico
   Terminal=false
   Type=Application
   Categories=Application;DIVA;Oracle;
   ```

4. To create the Configuration Utility shortcut, enter the following text and save the file as /usr/share/applications/diva-config-util.desktop:

   ```
   [Desktop Entry]
   Version=7.5
   Name=Config Utility
   Comment=Oracle DIVArchive Configuration Utility
   Exec=sh -c "cd /home/diva/DIVA/Program/ConfigUtility/bin/ &&
   ./configUtility.sh"
   Icon=/home/diva/DIVA/Program/ConfigUtility/bin/configUtility.ico
   Terminal=false
   Type=Application
   Categories=Application;DIVA;Oracle;
   ```

5. Use the following command to copy the shortcuts to the desktop after you have created both files:

   ```
   cp /usr/share/applications/{diva-control-gui.desktop,diva-config-util.desktop}
   /home/diva/Desktop
   ```

When you click each shortcut for the first time you may be asked if you trust the file. You must confirm them as being trusted files and they will be marked *trusted*.

Starting, Stopping, and Accessing DIVArchive in Linux

The following aliases become available after DIVArchive installation and are defined in `/home/diva/DIVA/Program/.divaenv.`:

```
alias DIVAgui="CurrDIR=`pwd`; cd /home/diva/DIVA/Program/GUI/bin; ./gui.sh; cd
${CurrDIR}"

alias DIVAconf="CurrDIR=`pwd`; cd /home/diva/DIVA/Program/ConfigUtility/bin;
./configUtility.sh; cd ${CurrDIR}"
```

> **Note:** All Linux paths, file names and command are case-sensitive.

Use the following procedure to start DIVArchive when running in a Linux environment:

1. Open a terminal console.

2. Change to the proper directory as follows:

    ```
    cd /home/diva/DIVA/std_linux
    ```

3. Start all DIVArchive services as follows:

    ```
    ./divaservice start-all
    ```

4. Open the Configuration Utility as follows (or use the Desktop shortcut):

    ```
    DIVAconf
    ```

    Use the following connection parameters:

    User Name
    Enter the database user name that was created.

    Password
    Enter the database user's associated password.

    SID
    Enter `lib5`

    Service Name
    Leave this field blank.

    IP Address
    Enter the IP address of the database host computer.

    Oracle Port
    Enter `1521`

5. Open the Control GUI as follows (or use the Desktop shortcut):

    ```
    DIVAgui
    ```

When shutting DIVArchive down, close the Control GUI and Configuration Utility. When they have closed, use the following command to stop all DIVArchive services:

```
./divaservice stop-all
```

You can use the following command to restart the services (if necessary) for any reason when they are already running:

```
./divaservice restart-all
```

3

Configuring DIVArchive Overview

This chapter describes a general overview of the DIVArchive configuration and includes the following information:

- Module Configuration Files
- DIVArchive Databases
- Environment Variables
- DIVArchive Configuration Utility Overview
  - Connecting to the DIVArchive Database
  - Disconnecting from the DIVArchive Database
- DIVArchive Control GUI Profiles and Passwords
- Changing the Database Logging Level
- Configuration Utility Tabs Overview
  - System Tab
  - Robots Tab
  - Disks Tab
  - Drives Tab
  - Tapes Tab
  - Sets, Groups & Media Mapping Tab
  - DIVAprotect Tab
  - Media Tab
  - Storage Plans Tab
  - Slots Tab
  - Manager Setting Tab

Module Configuration Files

Each DIVArchive software module has its own static configuration text file with parameters needed to launch that particular application. The files are typically denoted with the `.conf` file name extension. There are some DIVArchive modules that use an XML based file rather than a text file for their configuration and those will be noted where applicable.

Unlike older releases of DIVArchive that had stored these configuration files in the same folder as the application itself, DIVArchive 7.5 centralizes them to a dedicated `conf` subfolder under the DIVArchive Program Group.

The configuration files are typically updated with additional or changed settings in newer releases of the software. A new or patch release of DIVArchive will have the new releases of the `.conf` files appended with a `.ini` extension. For example, the new release of the DIVArchive Manager Configuration file will be named `manager.conf.ini`. You must remove the `.ini` extension after the installation is complete and the new configuration file updated.

Each configuration file can be opened and edited with any plain text editor (for example, Windows Notepad).

Any changes made to the configuration file of a DIVArchive software component requires that the component be shut down and then restarted for the changes to take effect. The exceptions to this are the Manager and Access Gateway options, both of which allow configuration changes to be reloaded while they are still running. There are codependencies between some applications in the DIVArchive platform, so other components may also need to be restarted for changes to take effect.

DIVArchive Databases

At the system level, settings that relate to the overall operation of each DIVArchive component and their interaction are configured and retained by an Oracle Database. This is commonly known (and will be referred to in this document) as the *DIVArchive Database* (or just simply as *the database*).

User modification of this database is performed through the DIVArchive Configuration Utility.

The DIVArchive Configuration Utility connects only to the database and does not necessarily require the DIVArchive Manager to be running. It is only intended for experienced users and caution should be exercised when altering settings using the utility. An incorrect setting can impede DIVArchive operations or prevent the DIVArchive Manager from starting successfully. Contact Oracle Support for assistance if you are unsure about making a particular change.

When launched, the DIVArchive Manager obtains the DIVArchive system configuration from the database. However, it does not poll the database for changes made through the Configuration Utility. Therefore, the database must be notified of any changes made. This is performed using the **Notify Manager** menu item in the Configuration Utility.

You can accomplish most changes to the configuration while the DIVArchive Manager is running. There are a small number of configuration changes that require a restart of the DIVArchive Manager to become effective. A full list of changes that can be made to the system configuration dynamically while the Manager is running is listed in Appendix B.

The Configuration Utility also does not dynamically poll the database for changes that are made when the Manager is running. In such cases, you click the **Update** button in the utility to refresh the information displayed from the database.

You can install the Configuration Utility on any computer that has TCP/IP connectivity to the database and a supported Java Runtime Environment installed. DIVArchive release 7.5 requires the Java Runtime Environment 64-bit (build `1.8.0_45-b14`), to be installed to launch the Configuration Utility successfully.

In some cases, a network firewall between the two can prevent a connection. For complete operation and functionality of the Configuration Utility, the *Oracle Listener Port* (typically `1521`) and the *DIVArchive Robot Manager Ports* (typically `8500` and higher) must be opened in the firewall. Full functionality of the Control GUI also requires that the *DIVArchive Manager Port* (typically `9000`) is open.

Metadata Database

The DIVArchive Metadata Database has very high performance and almost unlimited scalability. The Metadata Database should be treated with the same caution as the Oracle Database. It should be backed up at regular intervals through the DIVArchive Backup Service.

Oracle highly recommends that the Metadata Database files are stored on a RAID disk array. The Metadata Database should not be on a standard disk due to decreased performance and the real-time backup functionality that a RAID array offers the system.

Metadata Database files stored on a standard disk are vulnerable to data loss if a single disk failure until the information is replicated through the DIVArchive Backup Service. Storing the Metadata Database files on a RAID array isolates the data from this type of failure.

The information stored in the Oracle Database is already stored on a RAID-1 array and is not subject to data loss if a single disk fails.

Metadata Database Sizing

The following formula can be used as a rough guide to determine the minimum amount of disk space required to support the Complex Object Metadata Database:

```
(100+average_path_filename_size)*1.15*avg_num_component_files*num_objs
```

The following is a general example using the equation:

average_path_filename_size = 60
For example, `this/nested/subdir01/As_The_World_Turns_24fps_scenes1-10.avi`

avg_num_component_files = 200,000
The Average Number of Files and Folders within the complex object.

num_objs = 50,000
The number of complex objects to be archived.

In this example, *minimum* budgeting for a Metadata Database size of approximately 1.67 TB would be recommended.

When planning the system, you must allocate enough Metadata Database disk space to ensure for expected, or unexpected, growth of the environment. The same amount of disk space must also be allocated for the Metadata Database in all of the backup systems.

Environment Variables

Some DIVArchive software components may require defining one or more Windows operating system environment variables for those components to launch successfully.

An environmental variable allows the configured variable to be available to all programs rather than requiring it to be configured from the application each time it is

executed. This makes the variable independent of the application and therefore you do not need to manually insert or update the value when the application software is updated or modified.

A *User Environmental Variable* only applies to an individual Windows User Profile. A *System Environmental Variable* is applicable to all Windows User Profiles.

The following example illustrates how to configure the `DIVA_JAVA_HOME` environment variable on a Windows system. This variable defines the path of the Java Runtime Environment for DIVArchive applications on the Windows host. This particular parameter is required on any Windows computer that will run either the DIVArchive Configuration Utility or the DIVArchive Control GUI.

Use the following procedure to configure an environment variable:

1. Open the Windows Control Panel.

2. Double-click the **System** icon.

3. Click the **Advanced** tab.

4. Click the **Environment Variables** button.

5. Click the **New** button.

6. Enter the variable name in the *Variable name* field. In this example the name is `DIVA_JAVA_HOME`.

7. Enter the variable value in the *Variable value* field. This is the path (or other value) to use for the named variable. In this example the value is `C:\DIVA\java`.

8. Click **OK** to complete the process.

You have now defined the variable and it is displayed in the *System variables* list. The `DIVA_JAVA_HOME` environment variable is now accessible to all users (and applications) on the system and does not need to be defined each time an applications is executed.

DIVArchive Configuration Utility Overview

> **Caution:**   The Configuration Utility is intended only for experienced users. Incorrect or incomplete changes in the Configuration Utility can adversely affect DIVArchive operations (and possibly even delete data from the archive), or prevent the DIVArchive Manager from running. Contact Oracle Support for assistance if you are unsure about making desired changes.

The DIVArchive Configuration Utility primarily connects to the DIVArchive Database, and for some tasks, directly to the DIVArchive Robot Managers (if installed). After launching the utility you must first connect to the database to edit the DIVArchive system configuration. The Oracle user name and password for DIVArchive is arbitrary and can vary between installations. These details are documented in the DIVArchive Site Configuration for your site.

The utility can be installed and run on any computer with TCP/IP connectivity to the DIVArchive Database, Manager, and Robot Managers. Because it is a Java-based utility, a valid Java Runtime Environment (JRE) must also be installed on the host. For the Configuration Utility to launch, the `DIVA_JAVA_HOME` environmental variable in the host operating system must also be defined (see the previous section). This variable should match the absolute directory path to the JRE bin folder. For example `DIVA_`

HOME\Java (where DIVA_HOME is the chosen DIVArchive installation directory). The required release of Java is delivered with DIVArchive, and installed during the DIVArchive installation process.

Connecting to the DIVArchive Database

Use the following procedure to connect the Configuration Utility to the DIVArchive database:

1. Open the DIVArchive Configuration Utility.

2. Click **File**, and then **Connect**.

    Alternatively click the **Connect** icon just under the **File** menu.

3. Enter the following information in the fields on the DB Connection dialog box:

    User Name
    The database user name (typically diva).

    Password
    The Oracle password associated with the entered user name.

    S.I.D.
    The Oracle System Identifier (typically lib5).

    IP Address
    The IP address of the computer where the DIVArchive Database is installed.

    Oracle Port
    The Oracle Listener Port (typically 1521).

4. Click **OK** to connect to the database.

The connection status is displayed in the notification area at the bottom of the screen. If the connection fails an error message is generated, including an error code, and displayed in this area. Contact Oracle Support if you still cannot connect after attempting to resolve the error.

Disconnecting from the DIVArchive Database

Disconnect the Configuration Utility from the database when not in use. Use the following procedure to disconnect from the database:

1. Click **File**, and then **Disconnect**.

    Alternatively click the **Disconnect** icon just under the **File** menu.

2. Click **Yes** in the displayed Confirm dialog box.

3. Click **OK** in the displayed Disconnected dialog box.

The notification area at the bottom of the screen will display the *Not Connected* message.

Configuration Utility Frame Buttons

Each frame in the Configuration Utility includes a set of buttons that perform various functions as follows:

Plus (+)
This button is a plus sign (**+**). Clicking the button launches a dialog box to add an entry to the frame.

Edit
Highlighting a frame entry and clicking this button enables editing of the entry's properties.

Minus (-)
This button is a minus sign (**-**). Highlighting a frame entry and clicking this button will remove the entry from the frame. Entries with child dependencies cannot be removed.

Update
Clicking this button refreshes the associated frame content listing from the database.

DIVArchive Control GUI Profiles and Passwords

The DIVArchive Control GUI provides four fixed user profiles (Administrator, Operator, Advanced Operator, and User) to provide varying levels of access. The Administrator, Operator, and Advanced Operator profiles require a password that you can change using the Configuration Utility.

The difference between the Operator and Advanced Operator profiles are the **Insert** and **Eject** commands, which are only accessible from the Advanced Operator profile. You use the Operator profile during normal operations unless you are inserting or ejecting a tape.

There is no default password to log in to the Control GUI as an Administrator or Operator. You must assign an Administrator and Operator password in the Configuration Utility after DIVArchive installation is complete. Without an assigned password you are not permitted to switch to the respective profile in the Control GUI. If you attempt to switch to Administrator or Operator mode without first assigning a password to the profile, an error message is displayed notifying you that you must set a password. After you set the profile password in the Configuration Utility the first time, it no longer matters what you use for the *old password* when changing passwords.

Setting the Control GUI Administrator Password

Use the following procedure to set the Administrator password:

1. Open the Configuration Utility.

2. Connect to the DIVArchive Database.

3. Click the **Tools** menu item.

4. Click the **Change GUI Administrator Password** menu item (or use `Control+G`).

   This option is only available when the Configuration Utility is connected to the database.

5. Enter the following information in the appropriate fields on the Change GUI Administrator Password dialog box:

   Old Administrator Password
   Enter the old Administrator password in the *Old Administrator Password* field. You must leave this field blank the first time you set the Administrator password.

   New Password
   You enter the new password in the *New Password* field.

Confirm Password
You enter the new password again in the *Confirm Password* field.

6. Click **OK** to save the password.

Setting the Control GUI Operator Password

Use the following procedure to set the Operator password:

1. Open the Configuration Utility.

2. Connect to the DIVArchive Database.

3. Click the **Tools** menu item.

4. Click the **Change GUI Operator Password** menu item (or use `Control+G`).

   This option is only available when the Configuration Utility is connected to the database.

5. Enter the following information in the appropriate fields on the Change GUI Administrator Password dialog box:

   Administrator Password
   Enter the Administrator password in the *Administrator Password* field.

   New Operator Password
   You enter the new password in the *New Operator Password* field.

   Confirm Password
   You enter the new password again in the *Confirm Password* field.

6. Click **OK** to save the password.

Changing the Database Logging Level

Use the following procedure to change the Database Logging Level:

1. Open the Configuration Utility.

2. Connect to the DIVArchive Database.

3. Click the **Tools** menu item.

4. Click the **Change DB Logging Levels** menu item (or use `F12`).

   This option is only available when the Configuration Utility is connected to the database.

5. Use the menu lists to select the desired logging level for each package listed in the Change DB Logging Levels dialog box. Available levels are:

   FATAL
   When selected, this level displays very severe errors that may cause DIVArchive to terminate.

   ERROR
   When selected, this level displays errors that still allow DIVArchive to operate.

   WARN
   When selected, this level displays warning messages that are potentially harmful to operations.

INFO

When selected, this level displays informational messages about the progress of the operations.

TRACE

When selected, this level displays messages used to help debug the system.

6. Click **OK** to save your changes.

Configuration Utility Tabs Overview

The following sections describe a general overview of each tab in the Configuration Utility. Each tab includes multiple frames where you configure different aspects of the system.

To notify the Actors of any changes in the Actor configuration, click the **Notification** menu item, and then **Notify Actors** while connected to the Manager. The Actors must be running and connected to the Manager to receive the notifications.

System Tab

The System tab defines key parameters for your DIVArchive installation and is the starting point for creating your DIVArchive configuration.

You should create a drawing of the system components before entering details into the Configuration Utility. The drawing includes the data and control paths between components, how they interact with each other, established naming conventions for resources such as disks and tapes, and the workflow of the platform. Some parameters are difficult to change once they have dependencies from other configuration parameters in the database.

System Tab Frames

The **System** tab includes the following frames:

Production System Definitions

All DIVArchive installations have at least one production system. Additional production systems allow dedication of a particular Actor for specific destinations.

Sites

All installations have at least one site. Additional sites are optional and may be considered by the DIVArchive Manager for optimal resource allocation.

> **Note:** *Site Support* must be enabled in the DIVArchive Manager configuration, otherwise all sites will be considered equally.

Sources and Destinations

These define where DIVArchive archives from (Sources) and restores to (Destinations).

Actor Settings

These define Actor host definitions and logical functions. All installations have at least one Actor. *See Appendix A for Oracle DIVArchive options and licensing information*.

Transcoders

These define DIVArchive transcoders and analyzers. DIVArchive automatically selects the Actors either attached to a *Bitstream Flip Factory Transcoder* installation or

integrated with the *DIVAnalyze Harris QuiC* compressed file analysis software. DIVArchive allows a single transcoder to perform multiple transcodings. DIVArchive assigns additional ports as needed from the base port specified in the configuration. Therefore, a gap of one hundred between individual transcoder port settings is recommended to avoid port conflicts.

Actor Configuration in the Database

Except for the *Service Name* and *Port*, all Actor configuration settings are located in the Configuration Utility under the **Actor Advanced** and **Partial Restore Settings** tabs of the Actor frame of the **Systems** tab. Some settings are only available In *Engineering Mode* and are labeled with an **X** in the *Engineering Mode* column of the following tables.

Table 3–1   Advanced Actor Parameters

Name	Type	Minimum	Maximum	Engineering Mode	Default
DISABLE_DISK_ PREALLOCATION	Boolean			X	**Yes**
TAPE_TEST_UNIT_ READY_TIMEOUT (S)	Integer	60	1200		180
DO_NOT_CHECK_ OBJECT_NAME	Boolean				No
DO_NOT_CHECK_ CATEGORY	Boolean				No
SIMULATION	String			X	
SIMULATION_READING_ ERROR_RATE (%)	Integer	0	100	X	0
SIMULATION_WRITING_ ERROR_RATE (%)	Integer	0	100	X	0
SIMULATION_TAPE_SIZE (MB)	Integer	20	500000	X	300000
SEACHANGE_CHECK_ DELAY (MS)	Integer	0	10000	X	1000
PROFILE_READ_BLOCK_ SIZE (B)	Integer	1500	262144		1500
PROFILE_WRITE_BLOCK_ SIZE (B)	Integer	1500	262144		32768
QUANTEL_RENAME_ CLIPS	Boolean				**No**
QT_SELF-CONTAINED_ THRESHOLD (MB)	Integer	10	100		50
RENAME_TRANSCODED_ CLIPS	Boolean			X	
DIRECTORY_SERVER_ ENABLED	Boolean			X	**Yes**
DISK_FTP_PASSIVE_ MODE	Boolean				**No**
DISK_FTP_BLOCK_SIZE	Integer	1024	524288		32
DISK_FTP_SOCKET_ WINDOW_SIZE (B)	Integer	65536	10485760		65536

Table 3–2   Partial File Restore Parameters

Name	Type	Minimum	Maximum	Engineering Mode	Default
QT_IGNORE_START_TIMECODE	Boolean				**No**
QT_OMNEON_FIRST_FRAME_HANDLING	String				**Reset**
AVI_IGNORE_START_TIMECODE	Boolean				**No**
EVS_MXF_IGNORE_START_TIMECODE	Boolean				**No**
GXF_TIMECODE_REFERENCE	Integer	0	2		1
GXF_PROGRESSIVE_TIMECODE_TRANSLATION	Boolean				**No**
LXF_IGNORE_START_TIMECODE	String				**No**
MXF_PARTIAL_RESTORE_DICTIONARY_FILE	String				
MXF_TIMECODE_FROM_SOURCE_PACKAGE	Boolean				No
MXF_TIMECODE_VALUE_TO_SWITCH_PACKAGE	String				–1
MXF_ENFORCE_CLOSED_HEADER	String				**Yes**
MXF_RUN_IN_PROCESSOR	String				
MXF_IGNORE_START_TIMECODE	Boolean				**No**
MXF_USE_OMNEON_DARK_METADATA	Boolean				**No**
MXF_SERIALIZE_DEPTH_FIRST	Boolean				**No**
MXF_GENERATE_RANDOM_INDEX_PACK	Boolean				**Yes**
MXF_NUMBER_FRAMES_PER_BODY_PARTITION	Integer	50	500		250
MXF_UPDATE_TCTRACK_ORIGIN	Boolean				**No**
MXF_TOLERANCE_ON_TCOUT	Integer	0	250		0
MXF_DURATION_FROM_FOOTER	Boolean				**Yes**
MXF_MAX_QUEUE_SIZE	Integer	100	1000		200
SEACHANGE_IGNORE_START_TIMECODE	Boolean				**No**
MPEG2_TRANSPORT_STREAM_IGNORE_START_TIMECODE	Boolean				**No**

Table 3–2   (Cont.) Partial File Restore Parameters

Name	Type	Minimum	Maximum	Engineering Mode	Default
MPEG2_PROGRAM_ STREAM_IGNORE_ START_TIMECODE	Boolean				**No**

Robots Tab

All DIVArchive installations include the **Robots** tab, although not every installation necessarily has a library. This tab defines basic associations with the robotics software and hardware components.

Robots Tab Frames

The **Robots** tab includes the following frames:

Robot Managers

This frame defines to DIVArchive the connection parameters to each host running a DIVArchive Robot Manager Instance. *See Appendix A for Oracle DIVArchive options and licensing information*.

Libraries

Displays the tape or DVD libraries currently configured through one or more DIVArchive Robot Managers and their online status.

Media Compatibility

This frame maps the *Tape Media Type* defined in the **Tapes** tab, to the *Drive Types* defined in the **Drives** tab. Although you can manually remove entries in this frame, they can only be added or updated during a database synchronization with a Robot Manager.

Robot Managers-ACS

This frame associates each Robot Manager with an Automated Cartridge System (ACS) number. Although you can manually remove entries in this frame, they can only be added by performing a database synchronization with the specific Robot Manager.

Disks Tab

The **Disks** tab defines the physical disks that are to be used by DIVArchive, how they are grouped together for either permanent or cache storage, and how each disk is logically accessed by the Actors.

Disks Tab Frames

The **Disks** tab includes the following frames:

Arrays

An Array defines a logical association of disks in which one or more physical disks are assigned for use by DIVArchive. The *Array Name* is equivalent to the *Group Name* for tapes.

Disks

The symbolic name and location for each disk in your system, whether confined to a single host or shared between hosts. These disks are then assigned to Arrays.

Actor-Disk Connections

Configures how each disk is logically connected to each DIVArchive Actor, and how it is to be used. For shared disks accessible by more than one Actor, the disk connection must be declared for all Actors. *See Appendix A for Oracle DIVArchive options and licensing information.*

Cloud Accounts

Configures how each cloud account is logically connected to each DIVArchive Actor, and how it is to be used. *See Appendix A for Oracle DIVArchive options and licensing information.*

Drives Tab

The **Drives** tab is where the drives in your tape libraries are identified and configured for use with DIVArchive and its Actors. In some installations, a tape library and its drives may be shared with other applications. The configuration options enable you to disable any of the identified drives from DIVArchive use. *See Appendix A for Oracle DIVArchive options and licensing information.*

Drives Tab Frames

The **Drives** tab includes the following frames:

Drives

Displays the drives currently identified to DIVArchive in a database synchronization and their current status.

The Drive Edit dialog box enables editing the *Serial Number*, *Status* (online or offline), *Enabled Operations* (**Archive**, **Restore**, and so on), and *Used* (yes or no), information for a drive. This is useful if this information was not retrieved, or was entered improperly, during a SyncDB process. The firmware release for the drive is also displayed in a non-editable field. The firmware information is obtained from the Actors when they scan for tape drive devices. Other additional non-editable information is also displayed in this dialog box, and all of the information is displayed in the Drives frame.

Drive Properties

This displays the drive models currently configured for use with DIVArchive. Although you can manually remove entries in this frame, they can only be added by performing a database synchronization with a Robot Manager. *See Appendix A for Oracle DIVArchive options and licensing information.*

Actors-Drives

Indicates to DIVArchive which Actors have access to the drives configured in the Drives frame. *In this area associations can be added, edited, or deleted.*

Clicking the **+** button adds an existing Actor-Drive association. The Add New Row in Actors-Drives dialog box is displayed. Use the menu list to select the Actor, and then use the check boxes next to each drive to associate with the selected Actor.

Clicking the **Edit** button enables edit of an existing Actors-Drives association. The Edit Drives Entry dialog box is displayed. Make the required or desired updates and click **OK** to save the changes.

See Appendix A for Oracle DIVArchive options and licensing information.

Tapes Tab

The **Tapes** tab defines each *Tape Media Type* capacity in DIVArchive, and each individual tape's *write*, *repack* or *to be cleared* status. Tapes that do not contain any DIVArchive objects (that is, they are empty or are from another archive application in a shared library environment) and have been ejected from a DIVArchive managed library can also be deleted from the DIVArchive database from this tab.

Tapes Tab Frames

The **Tapes** tab includes the following frames:

Tape Properties

Displays the *Tape Media Types* and configuration parameters currently configured in DIVArchive after a library database synchronization.

In the Tape Properties frame, you can highlight an existing tape and click **Edit** to open the Tape Properties dialog box.

Empty Ejected Tapes

Displays the tapes that no longer have any DIVArchive content and have been ejected from an attached library. Clicking the **-** button permanently removes the selected tape from the DIVArchive Database.

Inserted Protected Tapes

When a tape is externalized, it is set to *Protected Mode* by DIVArchive. You must manually remove this state after reinsertion into the library if the tape is to have content written to it.

Tape States

A tape will appear in this frame if either the *Enable for Writing* or the *Enable for Repack* states is set to N. The *Enable for Writing* state can be automatically disabled by DIVArchive if it encounters an error during a read, write, or repack operation.

Click the **+** button in the Tape States frame to add a tape to the Tape States. Select the tape to add from the list in the displayed dialog box, and then click **OK** to add the tape.

The Tape States frame gives an overall indication of the reliability of your tape drives. Tapes appearing in this frame (if not manually inserted) indicates that either a read or write error occurred on that tape during DIVArchive operations. *If you have many tapes present here this may indicate an issue with one or more of your tape drives and should be promptly investigated*.

Sets, Groups & Media Mapping Tab

You use the **Sets, Groups & Media Mapping** tab to allocate tapes into pools for use by DIVArchive. The *Set ID* represents each media pool. The *Set ID* is typically used to distinguish different types of tape media. However, it may also be used to dedicate a specific set of tapes to specific groups.

A *Group* is a logical name for the storage of DIVArchive objects. Each group is assigned a *Set ID* of tapes to draw upon. Each group can only be assigned one *Set ID*. Several groups can share the same *Set ID*.

You can use the Configuration Utility to define the format of an array or group. The format is configured in the **Disks** and **Sets, Groups & Media Mapping** tabs for arrays and groups respectively. Alternatively, you can use the addGroup API call define a group or array and its format. The default format is AXF. This can also be achieved by

selecting **Legacy** in the Configuration Utility, or specify the corresponding value for the format using the API call.

Changing the format of an array is performed through the Edit Array Entry dialog box. Changing the format of a group is performed through the Edit Groups Entry dialog box. In either case, highlight the desired array or group and then click **Edit** to open the associated dialog box. Use the menu list to select the format (**Legacy** or **AXF**). Use the following procedure to change the format of an array or group:

1. Navigate to either the **Disks** (for arrays) or the **Sets, Groups & Media Mapping** (for groups) tab.

2. Highlight the desired array or group in the displayed list.

3. Click the **Edit** button at the top of the frame.

4. Use the *Tape Format* menu list to select the format for the selected array or group.

5. Click **OK** to complete the change.

Sets, Groups & Media Mapping Tab Frames

The **Sets, Groups & Media Mapping** tab includes the following frames:

Unused Tape Sets

Displays empty tapes that are recognized by DIVArchive and the library module where they are located. The *Set ID* of each tape can also be defined in this frame.

You can highlight an existing tape and click **Edit** to display the Edit Unused Tapes Sets Entry dialog box. When done editing the Tape Set click the **Refresh** button to refresh the list.

The *Unused Tape Sets* frame includes the following columns:

Barcode
These are tapes not currently in use by a group.

ACS
The ACS number is the specific library where the tape is located.

LSM
The LSM number is the specific library where the tape is located.

Media Type
This is the set's *Media Type*.

Set ID
This is the tape's *Set ID*. Click the desired tape to edit and then click the **Edit** button.

You can select multiple tapes by holding the **CTRL** key and clicking each tape. You select a range of tapes by holding the **SHIFT** key, click the first tape in the range, and then click the last tape in the range. Click **Edit** to open the Edit Multiple Rows dialog box.

Select the Set ID for the tape from the *Set ID* list. Only Set IDs that have already been created in the Groups window will be listed.

Setting the Set ID to 99 indicates that the DIVArchive is not to use the tape. This particularly applies to cleaning tapes installed in the library if they are reported to DIVArchive after a library audit. For example, a cleaning tape's typical barcode is CLNnnnn.

This also applies to some installations where DIVArchive shares its libraries with other applications. Tapes in use by other applications should also have their Set ID set to 99 to prevent DIVArchive from using them.

Groups

Adds, removes, or edits existing groups and each group's association with the tape pools defined in the *Unused Tapes Sets* frame. A group can only be removed when it no longer contains any DIVArchive objects.

Additional *Set IDs* for the *Unused Tape Sets* frame are only available after they are first created in a group. Tapes that should not be used by DIVArchive (for example, cleaning tapes) must be configured with a *Set ID* of 99.

The Groups frame includes the following columns:

Id

This is the library ID the group belongs to. This is automatically generated by the system and not editable.

Group Name

The name assigned to the group. These names will appear in the *MEDIA* list of an Archive request in the Control GUI.

Set ID

Default *Set ID* of each group is 1. You cannot assign tapes to additional *Set IDs* until after they are included in a group.

Description

This is an arbitrary description of the group.

Media Types

This is the tape *Media Types* currently in use by this group. This is updated automatically when a tape is assigned to the groups *Set ID* in the *Unused Tapes Sets* frame.

Tape Format

This is the format of the tape (**Legacy** or **AXF**).

Worse Fit Enable

By default, DIVArchive attempts to fill any tapes already assigned to a group before assigning an unused tape. The *Worst Fit* option attempts to span objects on as many tapes as possible.

Repack Reservation

This only applies if the *Worst Fit* option is enabled. It sets the number of unused tapes in the pool to reserve for tape repacking. All other groups that also use this group's *Set ID* must also have identical values.

VW

This column identifies whether *Verify Write* is on or off for the group.

Media Mapping

Media Mapping enables DIVArchive to automatically alter the specified media in an Archive request to another Disk Array, Tape Group or Storage Plan. In this way, you can alter the storage for Archive requests without requiring any changes in the archive initiator (automation or MAM system).

To edit an existing Media Map, highlight the desired mapping and then click **Edit** to edit the entry. Click **OK** to save your changes.

Click the **+** button in the Media Mapping frame to add a Media Mapping entry. Enter the *Name* for the mapping in the *Name* field. Use the menu lists to select the *From* (source), *Media to Map* to and *Storage Plan to Map* to, and then click **OK** to save the entry.

DIVAprotect Tab

The DIVAprotect settings are identified in the Configuration Utility's **DIVAprotect** tab as described in the following sections.

Configuration Frame

You set the main parameters in the Configuration frame as follows:

DB: Maximum possible history of Events in Months
Enter the number of months to retain DIVAprotect event history.

DB: Maximum possible number of Metrics
Enter the maximum number of DIVAprotect Metrics stored in the system. DIVAprotect will remove the oldest entries after this number is exceeded. This is completed through an automated database job that executes once per day, every day.

Manager: Size triggering Event Queue DB flush (nb events)
Enter the number of events collected in memory before saving them to the database.

Manager: Time delay triggering Event Queue DB flush (secs)
Enter the maximum interval for saving events to the database. If this interval is reached before the size triggering parameter is reached, the events will be saved to the database regardless of how many have been collected.

Event Definitions Frame

The Event Definitions panel displays the list of Event Definitions available for use in the metrics. Double-clicking an Event Definition or clicking **Open** will display a dialog box listing its associated parameters

Event Definitions are factory set and cannot be modified. Built-in metrics (`DIVAPROTECT*` metrics) cannot be edited and therefore do not appear in the Metric Definitions frame.

Metrics Definitions Frame

You double-click a Metric Definition to display an editing dialog box where the metric can be examined or modified. This has the same effect as selecting a metric in the list and clicking the **Edit** button.

The **+** and **-** buttons at the top of the frame enable adding or deleting a metric.

When adding a metric definition or editing an existing one, the Metric Definitions Properties dialog box is displayed. You can now enter or edit the following information as necessary:

Name
This is the name of the Metric Definition.

Description
This field enables you to enter a description of the Metric Definition that is displayed next to the Metric Name in the Metric Definitions panel. This description also appears

in the Control GUI when pausing your mouse over an entry of the Metric Definition list.

Enabled

Select this check box to enable the metric. Deselect it to disable the metric.

Collection Type

The *Collection Type* fields specify which event parameter (for example, Transfer Size) is collected as the data and the statistical computation performed on it (for example, Sum). Available statistics are as follows:

- Average

- Count

- Maximum

- Minimum

- Sum

- Weight-based Average

Weighted By

The *Weighted By* field specifies the divider parameter for Weight-Based Average collection (for example, Duration).

Collected Event

The *Collected Event* list specifies the events from which the collected event parameter is retrieved. The list will only display event types suitable for the parameter specified in the *Collection Type* second field. Event types that have no such parameter attached are absent from the listing.

Resource Type

The *Resource Type* field specifies which resource breaks down the data. For example, if you select **Drive Serial Number**, separate metrics will be generated for each drive. Use the menu list to select the resource type for the metric.

Interval

The *Interval* specifies the interval for metric calculation. For example, selecting **1 Day** will generate a metric each day (if corresponding data is available). The metric calculation is based on the associated events that occurred in the last 24 hours. Use the menu list to select the desired interval.

Default Events and Metrics Configuration

The following table identifies the default events and metrics that are internal to DIVArchive:

Table 3–3   Default Event and Metric Definitions

Event Field ID	Displayed Name	Is Aggregatable? Is Resource?	Is Collectable?	Date or Number	Quantifier
1	Event ID	No	Yes	Number	
2	Event Type	Yes	No		
3	Tape Type	Yes	No		
4	Tape Barcode	Yes	No		
5	Drive Type	Yes	No		

Table 3–3 (Cont.) Default Event and Metric Definitions

Event Field ID	Displayed Name	Is Aggregatable? Is Resource?	Is Collectable?	Date or Number	Quantifier
6	Drive Name	Yes	No		
7	Drive Serial Number	Yes	No		
8	Actor Name	Yes	No		
9	Object Name	Yes	No		
10	Object Category	Yes	No		
11	Object Instance	No	No		
12	Media	Yes	No		
13	Request ID	No	No		
14	Event End Time	No	No		
15	Event Duration	No	Yes	Number	Seconds
16	Transfer Size	No	Yes	Number	Bytes
17	Transfer Rate	No	Yes	Number	MB/Second
18	Transfer Error Rate	No	Yes	Number	Errors/GB
19	Error Code	Yes	No		
20	Error Message	No	No		
21	Disk Name	Yes	No		
22	Library Serial Number	Yes	No		
23	SD Name	Yes	No		
24	Transcoder Name / Analyzer Name	Yes	No		
25	Local DIVArchive System	Yes	No		
26	Number of Operations	No	Yes	Number	

Sample Metrics Definition

The following is a sample use case scenario:

You want to create your own metric for average duration of read and write operations on a tape in a DIVArchive system. You use the following procedure to create this metric:

1. Open the Configuration Utility.

2. Click the **DIVAprotect** tab.

3. Click the **+** button on the *Metrics Definitions* frame to open the Metric Definition dialog box.

4. Enter a unique name for the metric in the *Name* field.

5. Enter a description in the *Description* field.

6. Select the *Enabled* check box to enable the metric.

7. Set the *Collection Type* and *Weighted By* fields as appropriate using the menu lists.

   For example, if you select **Weight-Based Average** as the *Collection Type*, the *Weighted By* field is enabled. Because the *Weighted By* field is active, you are required to select a value to use to *weigh* the metric definition. In this case, the values for the *Weighted By* field are identical to the second *Collection Type* field.

8. Use the check boxes in the *Collected Event* area to select the events for collection.

9. Use the menu list to select the aggregation *Resource Type*.

10. Use the menu list to select the aggregation *Interval*.

11. Click **OK** to save your metric definition and complete the process.

Media Tab

The **Media** tab displays information (properties) of the media identified in the DIVArchive system. The display is for informational purposes and read only. You click the **Refresh** button to refresh the displayed list.

Storage Plans Tab

Caution: Misconfiguration of the Oracle DIVArchive Storage Plan Manager (SPM) may lead to unexpected and disastrous results! Minor changes can lead to catastrophic consequences. For example, the deletion of hundreds of thousands of instances on tape or database corruption. Without special training and familiarity with the product, it is recommended to contact Oracle Support before making any changes to SPM. Failure to do so may result in severe damage to the DIVArchive system or even permanent data loss.

The **Storage Plans** tab enables creation of simple and advanced rules for automated management and movement of content within the archive.

For detailed configuration information, see the *Oracle DIVArchive Storage Plan Manager (SPM) User's Guide* in the *Oracle DIVArchive 7.5 Additional Features documentation* library. *See Appendix A for Oracle DIVArchive options and licensing information*.

Storage Plans Tab Frames
The **Storage Plans** tab includes the following frames:

Storage Plans
Displays the Storage Plan names and definitions.

You click the **+** button in the *Storage Plans* frame to add a Storage Plan. Enter the Storage Plan name in the *Storage Plan Name* field. Use the menu lists to select whether to *Allow Last Instance Deletion*, *Please Specify Origin (Internal/External)* (this is typically **Internal**), and the *Group/Array Name* to associate with the Storage Plan. Click **OK** to save the changes.

You highlight a desired Storage Plan, and then click **Edit** to edit the selected Storage Plan. Click **OK** to save your changes.

Media Groups

Defines the tape groups or disk arrays to be allocated to slots, and if content deletion will be managed by the Storage Plan Manager.

You click the **+** button in the *Media Groups* frame to add a Media Group. Enter the *Medium Name* and *Storage Name* in the designated fields. Use the menu lists to select the *Group/Array Name*, *Watermarked*, and the *Disk Cleaning Strategy*. Click **OK** to save the changes.

You highlight a desired Media Group, and then click **Edit** to edit the selected Media Group.

Storage Plans

Displays the Storage Plan names and definitions.

You click the **+** button in the *Storage Plans* frame to add a Storage Plan. Enter the Storage Plan name in the *Storage Plan Name* field. Use the menu lists to select whether to *Allow Last Instance Deletion, Please Specify Origin (Internal/External)* (this is typically **Internal**), and the *Group/Array Name* to associate with the Storage Plan. Click **OK** to save the changes.

You highlight a desired Storage Plan, and then click **Edit** to edit the selected Storage Plan. Click **OK** to save your changes.

After you click **OK**, A warning dialog box appears asking whether you want to continue saving the changes. Click **Yes** to save the changes, or **No** to cancel.

Filters

This frame displays filter definitions related to the Storage Plan Objects. It enables performing actions on all or specific objects (based on object filters).

Slots Tab

This tab defines the Slots associated with the Storage Plans for the Storage Plan Manager. Slots define which tape groups or disk arrays are related to each storage plan, and the parameters for storage plan execution.

For detailed configuration information, see the *Oracle DIVArchive Storage Plan Manager (SPM) User's Guide* in the *Oracle DIVArchive 7.5 Additional Features documentation* library.

Slots Tab Frame

The **Slots** tab includes only one frame named Slots.

You click the **+** button in the *Slots* frame to add a Slot. Configure the Slot's parameters by entering the information desired for this slot, or using the menu lists to select the options. Click **OK** to save the changes.

You highlight a desired Slot, and then click **Edit** to edit the selected Slot. Click **OK** to save your changes.

The Slot Configuration screen serves two purposes; new slot configuration and editing an existing slot configuration. Both functions use the same dialog box. However, the information displayed in the dialog box is determined by whether a slot is being added, or an existing slot is being edited.

Manager Setting Tab

You use the **Manager Setting** tab to set several parameters related to the Media and the Metadata Database in the system.

Media Configuration

There are two settings to configure for the Media in the **Manager Setting** tab:

Media/Storage Plan Submission Delimiter

The object is assigned to a specific Storage Plan and saved to the specified media. The *Media Name* and the *SP Name* must be separated by the & delimiter.

Maximum Number of Records in `DP_OPERATIONS` Table

The maximum number of records maintained in the DP_OPERATIONS table in the database.

Metadata Database Configuration

You must configure the following three parameters in the **Manager Settings** tab to enable complex objects processing:

Complex Objects Metadata Database Location

Enter the full path to the Metadata Database files in this field.

Enable/Disable Database Backup Notification

Select the check box to enable the Metadata Database backup notifications, or deselect the check box to disable notifications.

Notifications must be enabled to receive DIVArchive Backup Service messages to the Control GUI. If this parameter remains disabled, there will be no notification of errors or warnings displayed in the Control GUI. The default is enabled (selected).

Enable/Disable Metadata Database Feature

Select the check box to enable the Metadata Database Backup feature. The default is disabled (deselected).

4

Start Configuring DIVArchive

This chapter describes initial DIVArchive configuration and includes the following information:

- Defining Production Sites and Other Sites
- Defining Sources and Destinations
- Source and Destination Overview
    - Source and Destination Configuration Limitations
    - Source Type
    - Connect Options
    - Root Path
    - Metasource
- Arrays and Disks
    - Defining an Array
    - Defining Disks
- Configuring Oracle Archive Cloud for DIVArchive

There are many interrelated components in a DIVArchive System. The following figure shows the basic configuration workflow.

The configuration of DIVArchive is hierarchical and top-level parameters such as Production Systems, Sites, Arrays, and Disks need to be configured before configuring other components such as DIVArchive Actors.

If you intend to modify an existing DIVArchive system, you must *always* start by backing up the existing DIVArchive installation, configuration files, and *especially* the DIVArchive Oracle and Metadata Databases.

Contact Oracle Support before making any modifications to your DIVArchive platform if you are unsure about any steps in the procedures, or require clarification.
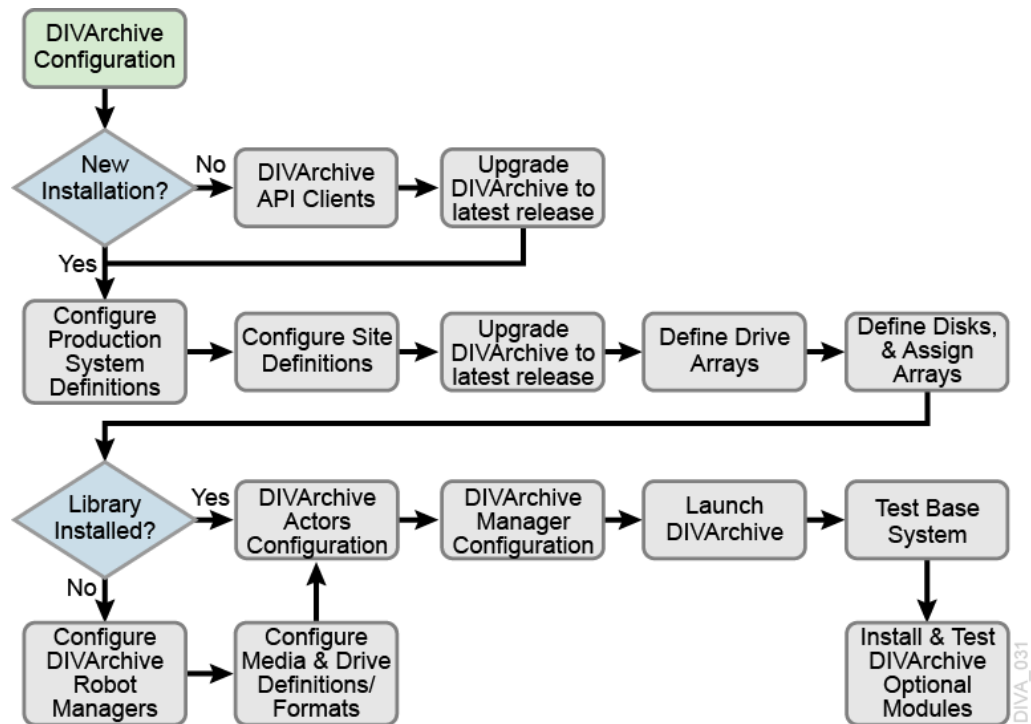
Defining Production Sites and Other Sites

A *Production System* is a logical group within DIVArchive that associates Actors to your sources and destinations. It enables splitting DIVArchive resources among different applications, or prioritizing specific functions over others within the platform. This is accomplished by assigning more DIVArchive Actors to one Production System over another. Although DIVArchive Actors cannot be shared between Production Systems, it is possible to share a source or destination. In this case, you can declare the specific source or destination more than one time, but each instance declared must have a unique name. *All installations must have at least one Production System defined*.

For applications requiring extremely high bandwidth, the Production System concept also enables you to dedicate an Actor to an individual source.

For example, you may have a Production System used for on-air transmission, and another Production System for offline editing. The Production System concept enables fine-tuning your resource allocation between the two systems based on the workflow and bandwidth requirements of each system.

A *Site* enables you to associate an Actor tape library or disk with a physical location. DIVArchive determines the most optimal use of resources during event execution. For example, a remote installation connected over a WAN that is used for disaster recovery purposes for the primary site. *All installations have at least one Site Definition*.

You identify Production Systems and Site Definitions under the Configuration Utility's **System** tab.

DIVA_021
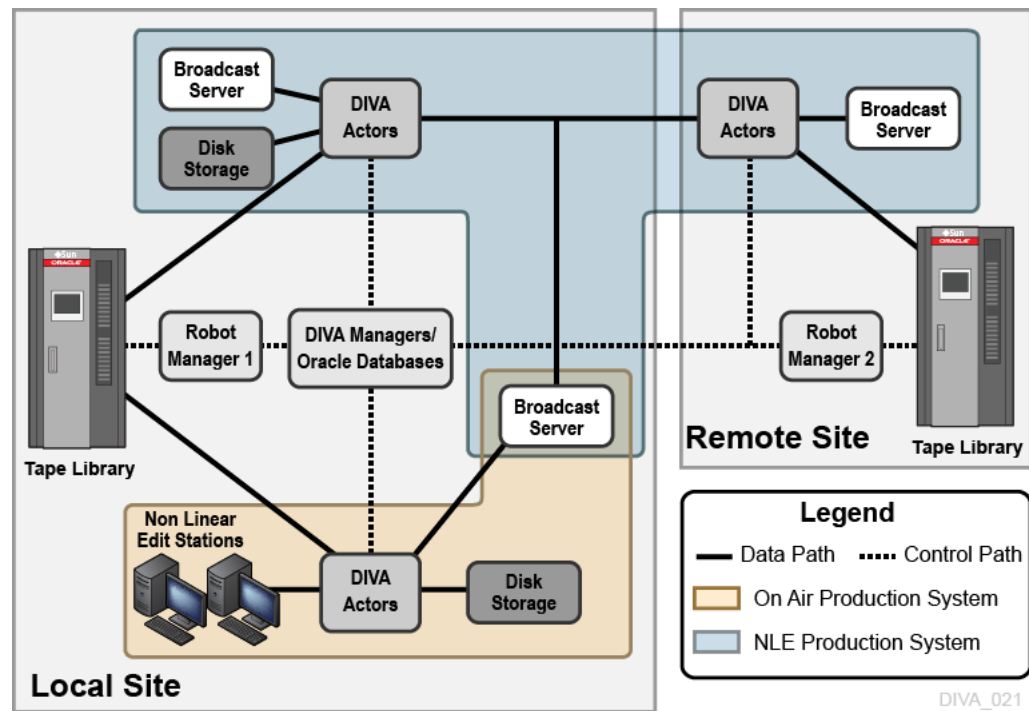
Defining Sources and Destinations

A *Source* is defined as any connected system that contains content intended to be transferred to DIVArchive. A *Destination* is defined as any connected system that requires content to be transferred to it from DIVArchive. Examples of both are Broadcast Video servers, FTP servers, and disk file systems.

These entries are defined in the *Sources and Destinations* frame of the **System** tab in the Configuration Utility. See Appendix C for details on specific settings for each type.

Use the following procedure to define a source or destination:

1. Open the Configuration Utility.

2. Click the **System** tab.

3. Click the **+** button at the top of the *Sources and Destinations* frame.

4. Enter the following information in the appropriate fields on the Sources and Destinations Entry dialog box:

   Source Name
   Enter a unique source name in this field. You can have multiple entries for the same physical source if they have different names. Multiple entries enables sharing the source with multiple Production Systems.

   IP Address
   Enter the IP Address or host name of the source. If you use host names they must be resolvable by the Actor host computers. Some *Source Types*, such as **Disk**, do not require an IP Address and can be left blank for those *Source Types*.

   Source Type
   Select a *Source Type* from the list. The Actor uses the *Source Type* to select the correct communications protocol for the device.

Prod. System
This field specifies this source's dedicated Production System.

Site
Select the *Site* location from the list. This field identifies the physical location of the source.

Connect Options
Enter any parameters required to connect to the source. For example, the user name and password.

Root Path
Enter the root directory path (if any) for accessing content on the source.

Max. Throughput (Mb/s)
This setting limits the total bandwidth used in transfers to or from this source. This is useful when the source does not natively provide any bandwidth throttling. This is typically set to a higher value than the source can provide (that is, unthrottled).

Max Accesses
Enter the maximum combined simultaneous read and write operations permitted to the source. For example, if this value is five and there are five requests active, any additional requests will be set to **Waiting for Resources** until at least one of the other requests has completed.

Max Read Accesses
The *Max Read Accesses* and *Max Write Accesses* enable fine tuning of transfers to and from the source. For example, you can allocate more restore operations than archive operations.

If the *Max Read Accesses* value is zero, this component is only used as a Source.

Max Write Accesses
The *Max Write Accesses* and *Max Read Accesses* enable fine tuning of transfers to and from the source. For example, you can allocate more restore operations than archive operations.

If the *Max Write Accesses* value is zero, this component is only used as a Destination.

First Utilization Date
This is a read-only field that displays the first date this component was put into service.

The remaining fields are related to Checksum Support and identified here for completeness. See the *Oracle DIVArchive Checksum Support User's Guide* in the *Oracle DIVArchive 7.5 Additional Features documentation* library for detailed information.

External Checksum Source
Select the appropriate option to identify whether the checksum source is external (*Yes*) or internal (*No*).

Checksum Type
Select the type of checksum (**MD5**, **SHA256**, and so on) in use from the list.

GC Mode
Select the Genuine Checksum (GC) mode from the list.

Verify Following Archive (VFA)
Select the check box to enable VFA on this source. Deselect the check box to disable VFA.

Verify Following Restore (VFR)
Select the check box to enable VFR on this source. Deselect the check box to disable VFR.

5. Click **OK** to save your configuration.

Source and Destination Overview

This section describes additional Source and Destination configuration and expands on the previous section.

Source and Destination Configuration Limitations

If you configure a Source/Destination with the `-rxml` option, a proper MD5 checksum is generated only if the object contains a single file. If the object contains more than one file, no checksums are generated. You typically use this configuration in DIVAnet configurations for restoring to a second DIVArchive system with checksums.

When this is the case, and an object that contains more than one file is restored, the resultant `.xml` Metadata file will not have checksums present. A Source/Destination on the receiving side configured with GC active will fail to archive the object because the checksum verification will fail (there is no matching checksum).

Source Type

The *Source Type* parameter of the Source or Destination definition establishes the specific protocol for the interface between the DIVArchive Actor and the target device or file system.

The source types supported by DIVArchive are predefined and selection is limited to the listing from the *Source Type* list.

Deprecated indicates that this is a software feature or function within DIVArchive that has subsequently been replaced by an improved feature set, however the feature or function is still supported by DIVArchive in the current release. The following is a list of source types available in the current DIVArchive release:

AM Communicator
Avid Archive Manager interface

TM Communicator DET Interface
AVID Unity interface using the Dynamically Extensible Transfer (DET) protocol.

TM Communicator DHM Interface
AVID Unity interface using the Data Handling Module (DHM) protocol.

CIFS
A disk source (that is assumed to be) visible to all Actors in the associated Production System. Buffered I/O will be used for transfers to and from this source.

Disk
A disk source (that is assumed to be) visible to all Actors in the associated Production System. Direct I/O will be used for transfers to and from this source.

EXPEDAT
DataExpedition (Expedat) Server

FTP_STANDARD
FTP protocol for RFC959 compliant servers.

LEITCH
This source type is deprecated, and is only provided for backward compatibility for older Leitch servers. Newer servers must use the **FTP_STANDARD** source type.

LOCAL
Represents a disk partition bound to a specific Actor.

MEDIAGRID
Omneon MediaGrid

METASOURCE
Used for accommodating multiple servers sharing the same online storage.

MSS
AVID MediaStream servers

OMNEON
Omneon Spectrum servers using unique FTP site commands

PDR
GVG Profile servers

QUANTEL_ISA
Quantel Q or sQ servers utilizing the Quantel Power Portal

QUANTEL_QCP
Older Quantel servers utilizing the QCP protocol

SEACHANGE_BMC
SeaChange Broadcast Media Clusters or Media servers using Vstream Streaming API

SEACHANGE_BML
SeaChange Broadcast Media libraries using FTP or CIFS protocols

SEACHANGE_FTP
SeaChange FTP support; *this Source Type is deprecated*

SFTP
Secure Shell (SSH) FTP protocol

SONY_HYPER_AGENT
For use with Sony Newbase FTP server

SIMULATION
For simulator platforms only. This creates simulated Sources and Destinations

One record is required for each Content Director that DIVArchive has to move data to and from. The following list identifies the Content Director attributes:

Table 4–1   Content Director Attributes

Attribute	Value	Example
IP Address	Leave this field empty	
Source Type	The desired source type	**MEDIAGRID**
Root Path	\\ContentDirector\file_system\clip.dir	\\10.30.0.200\cldev4\clip.dir \\mycontentdir\fs5\clip.dir

Connect Options

You define the *Default Quality Of Service* for DIVArchive transfer requests and additional protocol specific parameters (for example, user name and password) in the *Connect Options* field.

The *Connect Options* enable the DIVArchive Actor to establish a connection to the target device or disk file system. This applies to both requests submitted through the Control GUI or from third party archive initiators using the DIVArchive API.

See Appendix C for details on specific *Connect Options* for supported Broadcast Servers and file system types.

Root Path

You can explicitly specify the directory path at the request level. However, you can define the default directory path for the source, or a disk mount point for disk and local sources, in the *Root Path* field. The specified *Root Path* is appended before any *Files Path Root* specified in an archive, restore, or partial file Restore request unless the *Files Path Root* specified in the request is an absolute path.

The benefit of the *Root Path* approach is that you can specify the server's directories at the source and destination configuration level rather than at the request level. You can alter it without affecting commands issued from DIVArchive clients.

See Appendix C for details on the interaction of the *Files Path Root* and *Root Path* parameters.

Metasource

The optional *Metasource* enables DIVArchive to combine two or more existing Source/Destinations, which use common storage, or multiple Drop Folder Monitors, into a single Source/Destination configuration.

The *Metasource* feature offers load balancing and fault tolerance from within DIVArchive if an individual server or DFM fails. DIVArchive automatically attempts to use the next server (or DFM) in the *Metasource* configuration if the attached server (or DFM) is unavailable for a request.

Arrays and Disks

DIVArchive's disk management defines each physical disk, how it is attached (or mounted) to the system, and then groups the disks together to perform specific roles within the archive.

Defining an Array

The first step to disk management is to define an array. In DIVArchive an array is a logical grouping of one or more disks for the storage of DIVArchive objects. You define arrays in the *Arrays* frame of the **Disks** tab in the Configuration Utility.

Use the following procedure to define an array:

1. Open the Configuration Utility and connect to the DIVArchive Database.

2. Click the **Disks** tab.

3. Click the **+** button at the top of the *Arrays* frame to open the *Add new row in Arrays* dialog box.

4. Enter the following information in the fields on the dialog box:

   Id
   The array's ID is automatically generated by the system. The *Id* field is not editable.

   Array Name
   Enter a name for the array in the *Array Name* field. This is symbolic and typically represents the purpose of the stored objects.

   Description
   Enter a description for the array in the *Description* field. This is arbitrary and typically denotes the array's function.

   Format
   Select the array format from the list. Options are **AXF** and **Legacy**. The **AXF** format is required for complex objects.

   Max Allowed Disk Space For Repack (%)
   Enter the percentage of disk space available for use by repack requests.

   Verify Write (VW)
   You select whether to enable Verify Write from the list. Verify Write is not compatible with complex objects.

   Default Checksum Type
   The **MD5** algorithm is the *Default Checksum Type*. This field is not editable in this dialog box.

5. Click **OK** to save the array configuration.

You highlight a desired array and click **Edit** on the top of the *Arrays* frame to edit an existing array. When you finish making changes, click **OK** to save your changes.

> **Note:** Existing arrays cannot be edited or removed while they are referenced by a disk, or contain DIVArchive objects.

Defining Disks

Next you define the physical disks that are going to be used by DIVArchive, and assign them to arrays based on their intended function. You configure disks in the *Disks* frame of the **Disks** tab.

Each configured disk represents a distinct physical volume. Logical associations of disks to DIVArchive are performed in the *Actor-Disk Connections* frame

You can assign each disk declaration in this frame to only one Array. If you intend to share a physical disk between two or more arrays, you can declare the disk multiple times, but each declaration must have a unique name.

Defining how the disks are actually interfaced to DIVArchive is performed in the *Actor-Disk Connections* frame of the **Disks** tab.

Use the following procedure to define a disk:

1. Open the Configuration Utility and connect to the DIVArchive Database.

2. Click the **Disks** tab.

3. Click the + button at the top of the Disks frame to open the *Add new row in Disks* dialog box.

4. Enter the following information in the fields on the dialog box:

    Disk Name
    Enter a symbolic name for the disk in the *Disk Name* field. Oracle recommends that the name describes its function or its location.

    Array
    Assign the disk to an array selected from the menu list. Only arrays configured in the *Arrays* frame are listed.

    Site
    Select the *Site* that defines the location of this disk. This parameter is taken into consideration by DIVArchive for optimum allocation of disk resources in the array if the *Site Selection* parameter is enabled in the DIVArchive Manager configuration file

    Status
    Set the current status of the disk using the list (**ONLINE** or **OFFLINE**). **OFFLINE** indicates that the disk is offline and not to be used. During DIVArchive operations, the status may be set **Offline** by DIVArchive if an unexpected disk I/O error occurs.

    Min. Free Space, MB
    You set the minimum free space of the disk in this field. When the remaining free space reaches this amount, DIVArchive considers the disk full. Use this setting on disks that are shared with other applications, or with file systems that suffer poor or degraded performance when approaching 100% capacity.

    Verify Write (VW)
    You select whether to enable Verify Write from the list. Verify Write is not compatible with complex objects.

    Default Checksum Type
    The **MD5** algorithm is the *Default Checksum Type*. This field is not editable in this dialog box.

5. Click **OK** to save the disk configuration.

You highlight a desired disk and click **Edit** on the top of the *Disk*s frame to edit an existing disk. When you finish making changes, click **OK** to save your changes.

Configuring Oracle Archive Cloud for DIVArchive

You can use your Oracle Archive Cloud account for operations in DIVArchive. Any disks added to cloud arrays are considered cloud disks.

You must create an array with a storage class of **ARCHIVE** or **STANDARD**. Content stored using the **ARCHIVE** storage class requires a minimum of four hours retrieval time. Content stored using the **STANDARD** storage class is immediately available for retrieval. The **NONE** storage class is reserved for non-cloud arrays.

In the *Actor-Disk Connections* frame a cloud disk must be designated with the new cloud interface type, and must have a mount point that points to a cloud account. When you select a cloud interface during the creation or modification of an Actor-Disk connection, the corresponding *Mount Point* updates with a list of all cloud accounts and you can select the appropriate cloud account from the list.

> **Note:** The cloud account name must not contain a colon and cannot start with `cifs`.

Cloud accounts are defined in the *Cloud Accounts* frame. You must specify a unique *Account Name*, *Login*, *Password*, *URL*, *Service Name*, and *Identity Domain*. These five parameters are configured when you creates an account through Oracle's Cloud Configuration site (`https://cloud.oracle.com`). The parameter controlling the number of threads used by an Actor in writing content to the cloud is also configurable in the Cloud Accounts frame.

Actor-Disk connections including a disk belonging to an array with an **ARCHIVE** or **STANDARD** storage class must have the *Cloud Interface* setting and a *Mount Point* pointing to a cloud account name.

The Actor-Disk connection interface must be the same for all disks of a specific array, that is, either all cloud interfaces or all non-cloud interfaces.

Using an Actor-Disk connection with a cloud interface must be identified as **STORAGE ONLY**. Cache and Nearline are not supported for cloud content.

The ability of an Actor to archive or restore from the cloud is configurable in the **Actor Settings** tab of the *Actor Settings Entry* dialog box. Only Actors with *Cloud Archive* enabled will be used for *transfers to* the Oracle Archive Cloud. Only Actors with *Cloud Restore* enabled will be used for *transfers from* the Oracle Archive Cloud.

For example, when you are copying from tape to the cloud, the Manager will only use Actors configured for Cloud Archive. When you are copying from the cloud to tape, the Manager will only use Actors configured for Cloud Restore. Both Cloud Archive and Cloud Restore require **Direct Archive** and **Direct Restore** respectively. These settings are enabled when the corresponding cloud settings are configured.

5

Configuring DIVArchive Robot Managers

This chapter describes DIVArchive Robot Manager configuration and includes the following information:

- Configuration Overview
- SCSI Connected Libraries
  - Fibre Channel Host Bus Adapter (HBA) and SCSI Persistent Binding
  - Determining the SCSI Library Connection
- Sony ODA Drives
- Configuring Direct Attached SCSI Libraries
- Configuring ACSLS Attached Libraries
- Configuring Sony PetaServe Libraries
- Configuring ADIC Libraries with SDLC
- Configuring Simulated Libraries (for DIVArchive Simulators)
- Robot Manager Command Options
  - Installing and Uninstalling the Robot Manager Services in Windows
  - Installing and Uninstalling the Robot Manager Services in Linux
  - Robot Manager Service Management Functions
- Testing the Robot Manager Library Interface
  - Starting, Stopping, and Restarting the Robot Manager
  - Testing the Robot Manager Library Control
- Configuring the Robot Manager at the System Level
- Logging Robot Manager Activity
  - Managing SCSI Errors
- Configuring Media and Drive Types
  - `SCSI_drive_types` and `ACSLS_drive_types`
  - `SCSI_tape_types` and `ACSLS_tape_types`
  - `ADIC_media_types`
- Defining Tape Capacity and Block Sizes
- Synchronizing Media Types with the Database

- Synchronizing Drive Types with the Database

- Synchronizing Media and Drive Compatibility with the Database

- Synchronizing the Library Drive List with the Database

- Manually Identifying Drive Serial Numbers

- Creating Tape Groups

- Creating Tape Sets

- Remapping Media

Configuration Overview

The DIVArchive Robot Manager on Windows platforms runs as a Windows service and is launched automatically with Windows. *See Appendix A for DIVArchive options and licensing information.*

You configure the type of interface a specific library in a static configuration file. The file name is `robotmanager.conf` and is located in the `DIVA_HOME\Program\conf\robot_manager` folder on the computer where the DIVArchive Robot Manager is installed. In a new installation (or upgrade) the file is provided with a `.ini` extension. You must remove the extension for it to be acknowledged by the Robot Manager. You must copy the file, remove the `.ini` extension, and then edit the new file.

Since many different types of libraries and connections are supported, not all sections of the configuration file will be relevant to your particular installation. Also, some parameters are specific to the operating system where the Robot Manager is installed. Therefore, some settings in the configuration file are initially commented out (that is, they have # in front of the parameter). This indicates to the Robot Manager to ignore the setting. For the setting to be taken into account the # must be removed.

The following figure outlines the steps for configuring the robotics to be controlled by DIVArchive:
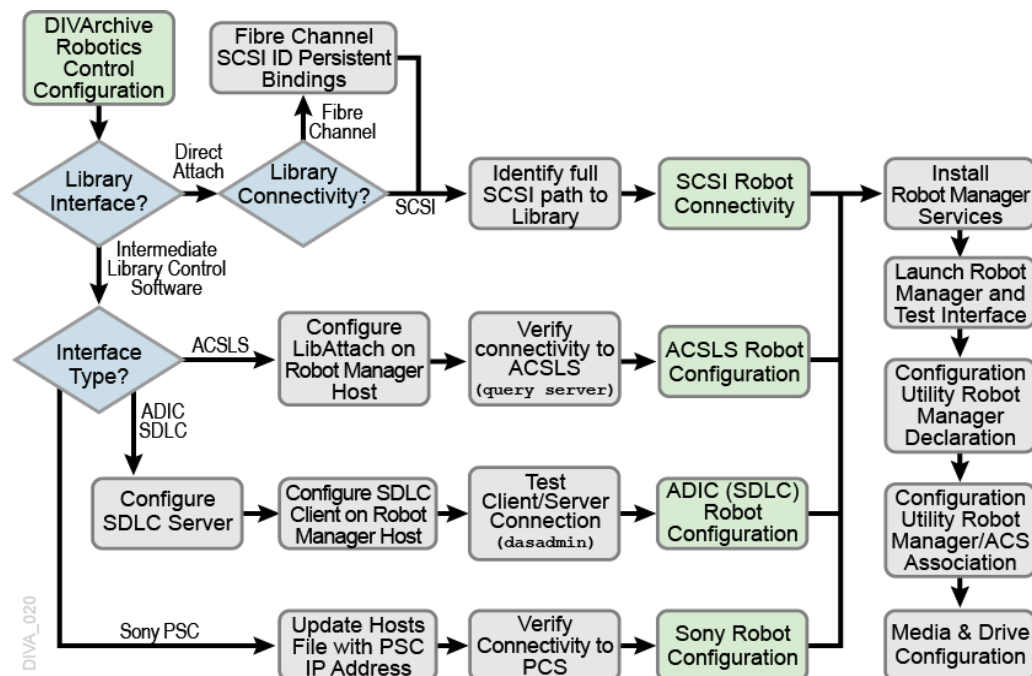
SCSI Connected Libraries

For directly attached SCSI controlled libraries, you must configure and correctly identify the *SCSI ID* controlling the library, and enter this value into the `RM_SCSI_DEVICE_LSM` parameter in the Robot Manager configuration file. Before changing the configuration, you must understand several concepts as described in the following sections.

The `robotmanager.conf` configuration file includes the following main parameters:

RM_SCSI_MOVEMEDIUM_TIMEOUT
Robot SCSI uses the `MOVE MEDIUM` SCSI command during mount, dismount, enter, and eject requests. The value of `RM_SCSI_MOVEMEDIUM_TIMEOUT` is indicated in minutes, and the default timeout is fifteen minutes for the communication between the library and the robot manager.

Some libraries, like Spectra T950, may require more time to be able to complete a `MOVE MEDIUM` request and you should set this parameter value accordingly.

RM_SCSI_EJECT_USEGLOBALLOCK
You must set this parameter to one if you want the SCSI Robot Manager `Eject` calls obtain the lock number of the LSM and hold that lock until all associated tapes to be ejected have completed the ejection process. When all tape ejections are complete, the call unlocks the drive and proceeds on to the next drive. The default setting is zero.

Fibre Channel Host Bus Adapter (HBA) and SCSI Persistent Binding

Most installations use Fibre Channel (FC) rather than native SCSI to interface to the library (typically over a SAN). In these instances, the FC HBA in the DIVArchive Robot Manager host presents the *World Wide Name* of the library interface as a *SCSI ID*. By default, most HBAs automatically map these to a *SCSI ID* for the host operating system to access. This presents a problem if a device is added or removed on the SAN because it could alter the *SCSI ID* of the library by the HBA, and automatically remap the existing devices. Disable the *Automap* feature to avoid this issue and use *Persistent Bindings* instead. This feature allows the SCSI mapping of the library to remain consistent between host restarts, and from the advent of any addition or removal of devices on the SAN.

If the library controller or the HBA in the DIVArchive Robot Manager host is changed, this might alter the library's SCSI Persistent Bindings to the host operating system. This requires the Persistent Binding for the library to be reconfigured in the HBA configuration software on the DIVArchive Robot Manager computer.

Determining the SCSI Library Connection
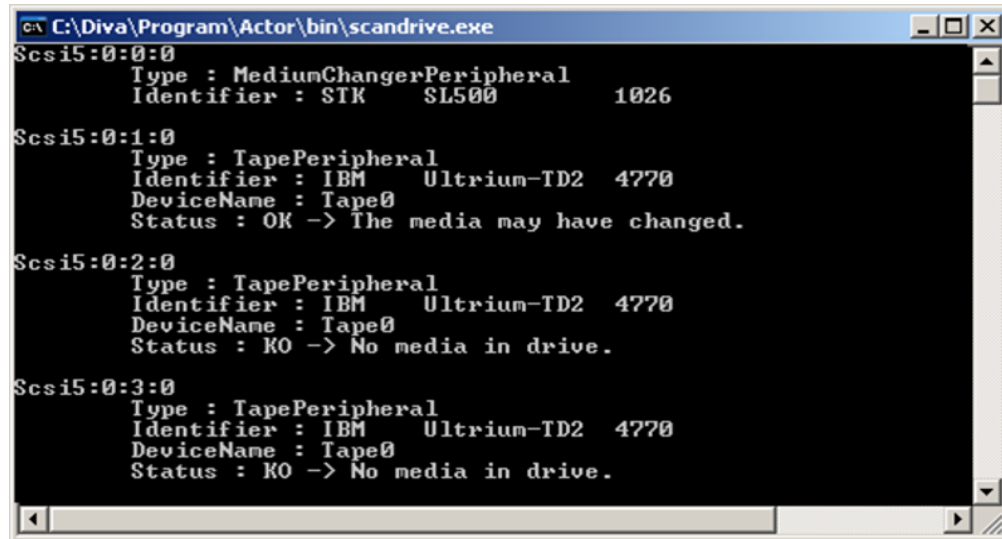
For most SCSI interface libraries the DIVArchive Robot Manager communicates with the library directly over the SCSI hardware layer and does not require the traditional Windows driver interface. The exceptions are IBM and HP libraries.

For all other libraries it is *essential* that no library driver be loaded for the library interface. If a driver is loaded, the DIVArchive Robot Manager will be unable to communicate with the library. In this case, if your library does not appear in Windows Device Manager as an *Unknown Medium Changer*, the Robot Manager will be unable to communicate correctly with the robotics.

If you cannot locate a specific library in the Scandrive Utility (see the following), but that library is visible in your HBA, then the library has likely been disabled in the

Windows Device Manager (denoted by an X over the device icon). You must re-enable the device for it to appear in the Scandrive Utility.

For Windows, you can determine the `RM_SCSI_DEVICE_LSM(n)` settings for the DIVArchive Robot Manager using the `scandrive.exe` utility. The utility is located in the `%DIVA_HOME%\Actor\bin` directory. The utility automatically reports all devices located in the Windows SCSI hardware tree in the registry and their corresponding Port, Bus, Target, and Logical Unit Numbers (LUN).

```
C:\Diva\Program\Actor\bin\scandrive.exe                              _ □ X
Scsi5:0:0:0
        Type : MediumChangerPeripheral
        Identifier : STK    SL500          1026

Scsi5:0:1:0
        Type : TapePeripheral
        Identifier : IBM    Ultrium-TD2  4770
        DeviceName : Tape0
        Status : OK -> The media may have changed.

Scsi5:0:2:0
        Type : TapePeripheral
        Identifier : IBM    Ultrium-TD2  4770
        DeviceName : Tape0
        Status : KO -> No media in drive.

Scsi5:0:3:0
        Type : TapePeripheral
        Identifier : IBM    Ultrium-TD2  4770
        DeviceName : Tape0
        Status : KO -> No media in drive.
```

The utility reports the SCSI Device ID of the library in the format `ScsiP:B:T:L` (see the previous figure), where `P` is the port number, `B` is the bus number, `T` is the target number, and `L` is the Logical Unit Number.

The `Type` section of the utility's output refers to that peripheral's class (HDD, CDROM, and so on). A tape library will be reported as a *Medium Changer Peripheral*, and the `Identifier` for each corresponding device reported should match the model number of the library itself (for example, SL500). You can then enter the full SCSI path reported for each library into the `RM_SCSI_DEVICE_LSM(n)` settings in the `robotmanager.conf` file.

Sony ODA Drives

Sony released a new generation of ODA drives; the ODS-280F (Fibre Channel) and ODS-280U (USB). DIVArchive has only been tested with the Fibre Channel type. The drives are twice as fast as the Gen1 drives. The ODS-280U has not been qualified for use with DIVArchive.

A new cartridge type is also available for this drive, the ODC3300R. This is a WORM drive with a 3.3 TB capacity.

Gen2 drives can read content written on Gen1 media with Gen1 drives. DIVArchive does not support the READ-ONLY media-drive compatibility. Oracle recommends isolating Gen1 media from Gen2 media in the configuration (because there is no cross-generation compatibility) and there must be at least one Gen1 drive in a library containing Gen1 cartridges.

DIVArchive supports Sony ODA ODS-D55U and ODS-D77F drives only in the Windows environment. These are Blu-ray Optical Drives and the media is WORM media using a UDF format. Only AXF formatted objects can be written to the discs.

The drives are controlled by the Robot Manager and the media is viewed as a Tape Cartridge.

In the Windows Device Manager these drives will be shown as *Unknown Medium Changer* under the *Medium Changer* section because there are no device drivers for them. The drive itself will also appear as an *Optical SCSI Device* with the make and model number under the *Disk Drives* section.

There are seven different types of disc media available for use with the Sony Optical Drives as follows:

SONY-ODC300R
293,265,408 KB capacity

SONY-ODC300RE
293,265,408 KB capacity

SONY-ODC600R
586,530,816 KB capacity

SONY-ODC600RE
586,530,816 KB capacity

SONY-ODC1200RE
1,173,086,208 KB capacity

SONY-ODC1500R
1,500,020,736 KB capacity

SONY-ODC3300R
3,222,717,696 KB capacity

The disc types are identified in the `scsi_tape_types.ini` file (described in the following section).

> **Note:** You must configure the drive settings *before* configuring DIVArchive. The recommended parity setting is **PARITY ON**.

You can view the drive specifics using the Optical Disc Archive Utility. This utility enables viewing of device logs, and viewing and changing drive settings.

To change the drive settings, click the **Setting** tab in the Optical Disc Archive Utility. Oracle recommends leaving the *Default Volume Type* set to **PARITY ON**, and to use the default settings for the remaining items.

Click the **Media** item under the **Drive** navigation tree to view information about the media in a drive.

You click the **Write-protect** button to write-protect a drive. Once an Optical Disc is write-protected, you can no longer write objects to the device. However they are still retrievable.

Configuration File Adjustments

You must change several parameters in the `scsi_drive_types.ini` configuration file to use these optical drives.

In the `robotmanager.conf` configuration file, under the SCSI module specific options, the serial number must be identified. You can find the serial number in the `RM_SCSI_`

`DEVICE_LSM(n)` parameter line. For example, `RM_SCSI_DEVICE_LSM(0)=00001003`, where `(0)` is the LSM number, and `00001003` is the serial number. You must identify the serial number for all listed devices (`LSM(0)`, `LSM(1)`, `LSM(2)`, and so on).

In the `scsi_drive_types.ini` file, the *drive types* must be uncommented (remove the #). For example, remove the # from in front of the line that reads `#601 0x00 0x00 SONY-ODS-D77F 600 601 602 603 604 605` to use your D77F drive as shown. The `TransportDomain` and `TransportType` are obtained automatically and not used in the configuration, so you must leave these set to `0x00` as shown in this example.

```
#-------------------------------------------------------------
# If the SCSI Robot Manager is connected to a SONY ODA library
# UNCOMMENT ALL LINES IN THE FOLLOWING PART
#-----
#TypeID TransportDomain TransportType TypeName CompatibleTapeTypes
#--------------------------------------------------------------------
#600 0x00 0x00   SONY-ODS-D55U    600 601 602 603 604 605
601 0x00 0x00    SONY-ODS-D77F    600 601 602 603 604 605
```

Also, in the `scsi_tape_types.ini` file, uncomment all of the *disc types* listed as shown in the following example. The `R` or `RE` after the disc number indicates whether the disc is *Write Once* (`R`) or *Rewritable* (`RE`). This indicator is used because the barcode does not contain the video type as in normal tape barcodes.

```
#--------------------------------------------------------------
# If the SCSI Robot Manager is connected to SONY ODA library,
# UNCOMMENT ALL LINES IN THE FOLLOWING PART
#-----
#TypeID TransportDomain TransportType TypeName CompatibleDriveTypes
#--------------------------------------------------------------
600 0x00 0x00   SONY-ODC300R    600 601
601 0x00 0x00   SONY-ODC300RE   600 601
602 0x00 0x00   SONY-ODC600R    600 601
603 0x00 0x00   SONY-ODC600RE   600 601
604 0x00 0x00   SONY-ODC1200RE  600 601
605 0x00 0x00   SONY-ODC1500R   600 601
```

Configuration Utility Settings and Information

You must configure the following settings in the DIVArchive Configuration Utility:

Drives Tab
Set the *Drive Properties* to `64 KB`. The serial number comes from the Robot Manager and the firmware release number comes from the drive.

Tapes Tab
The *Tape Properties* frame displays all of the enabled *Tape Types* from the `scsi_tape_types.ini` file.

Control GUI Settings and Information

The *Optical Drives* and *Discs* are displayed in the DIVArchive Control GUI on the **Drives** tab as *Tape Drives* and *Tapes* respectively.

Repack of the discs and deletion of objects is available. However, the space is not recoverable. When trying to repack the disc, the normal Repack dialog box is displayed, but there is a warning that the space is non-recoverable. Due to this limitation of the discs, auto-repack has been disabled for these drives and discs.

Additional Information

Additional information related to the use of the Optical Drives and Discs includes the following:

- Because Write-Once media must be finalized, zero remaining space will be reported to the Manager.

- Objects are spanned when there is 100 MB of space remaining. This is so that there is space left for the disc to be finalized. Once an object is spanned, the disc is considered full and is automatically finalized.

- The Actor will auto-finalize the discs when there is 500 MB of space remaining unless an object was spanned. However you can manually finalize the disc through the Optical Disc Archive Utility.

- If a drive is manually mounted and viewed in the Windows Explorer, the display will show the individual files on the disc. Each file name will begin with a numeric value at the beginning that identifies the object's location on the tape.

Configuring Direct Attached SCSI Libraries

A *Direct Attached Library* is directly connected to the DIVArchive Robot Manager host computer either through a native SCSI interface and SCSI HBA, or through a SCSI over Fibre Channel connection and Fibre Channel HBA (either directly or through a SAN).

In either case, the DIVArchive Robot Manager uses its own DIVArchive provided driver (`SCSI_Robot.dll` in Windows or `libSCSI_Robot.so` in Linux) to directly interface with the library without the need for intermediate library management software. For this type of SCSI attached library, you must uncomment the entries (in the following sections) and configure them in the `robotmanager.conf` file. Library *Drive Models* and *Tape Types* parameters are located in other configuration files.

Common Settings for SCSI-based Libraries

The following are typical settings for the SCSI-based libraries:

Robot Manager Common Options

Uncomment only the `RM_MODULE=SCSI_Robot.dll` in the Windows environment.

SCSI Device Parameters

The following table identifies common SCSI device parameters.

Table 5–1   SCSI Device Parameters

Parameter	Parameter Type	Description	Default
SERVICE_NAME	Name	The display name of the Robot Manager Windows Service. You must set this variable if multiple Robot Managers are installed on the same server. If this variable is used, the *Service Name* will be `DivaRbt-SERVICE_NAME`. If this variable is not set, the *Service Name* will revert to just `DivaRbt`.	Uncommented

Table 5–1   (Cont.) SCSI Device Parameters

Parameter	Parameter Type	Description	Default
RM_PORT	TCP port number	The TCP port that the DIVArchive Robot Manager listens on for incoming requests. This value must be unique if there are multiple DIVArchive Robot Managers running on a single host computer. This is typically, TCP port 8500 and greater.	8500
RM_ACS	Number	The Automated Cartridge System (ACS) controlled by the DIVArchive Robot Manager module. This value will appear in the *Robot Manager/ACS Association List* in the Configuration Utility for this Robot Manager after database synchronization	0

SCSI Module Parameters

The following table identifies common SCSI module parameters. See Determining the SCSI Library Connection for parameter details.

Table 5–2   SCSI Module Parameters

Module Parameter	Operating System	Description	Values
RM_SCSI_DEVICE_LSM0	Windows	This specifies the SCSI target of the library as it identified by the host operating system.	ScsiP:B:T: L
RM_SCSI_DEVICE_LSM1	Windows	This setting is specific to a StorageTek dual L1400M library with a Pass Through Port (PTP), and specifies the SCSI target of the 2nd frame (LSM). Although this type of library configuration can be addressed using only the RM_SCSI_DEVICE_LSM0 connection, DIVArchive manages this type of library more effectively when both frames are specified. DIVArchive also manages the PTP in this case.	ScsiP:B:T: L

Additional Settings for Qualstar and ADIC (Return Media Id Enabled) Libraries

The following table identifies additional settings specifically related to Qualstar and ADIC libraries.

Table 5–3   Additional Settings for Qualstar and ADIC Libraries

Parameter	Parameter Type	Description	Default
RM_SCSI_ENABLE_MEDIA_TYPE_DETECTION	0 (disabled) 1 (enabled)	This option specifies whether the *Tape Media Type* may be embedded in the barcode labels returned by the library. A barcode label is usually a six digit string. However, if this feature is enabled, these libraries return eight digit strings. The two additional characters are not part of the actual label, but define the media type. If this value is enabled, the Robot Manager will determine the *Tape Media Type* by using the label and removing the two additional characters from the label.	0
RM_SCSI_MEDIA_TYPE_LEFT_DETECTION	0 (disabled) 1 (enabled)	This option indicates to the Robot Manager the location of the extra two characters if the RM_SCSI_ENABLE_ MEDIA_TYPE_ DETECTION option is enabled. Depending on the specific library, the additional two characters may be on the right or on the left of the label. If the media type information is on the left, set this value to 1 (enabled), otherwise, set it to 0 (disabled).	1

Additional Settings for DVD Jukeboxes (ASACA or Plasmon)

The primary difference between a DVD Jukebox and a regular Tape Library is that the former does not implement the **READ VOLUME ELEMENT ADDRESS SCSI** command. The media in the Jukebox is identified by their element address and not their volume tag or label. Conventional tape libraries allow the element location of a barcode ID to be provided by the library. DVD Jukeboxes do not provide this functionality. Due to this limitation of the DVD Jukeboxes, the DIVArchive Robot Manager for this library instance must maintain its own local media database of element addresses.

> **Tip:** Intermediate Windows drivers (if loaded) for the Jukebox can cause an issue with the control of the Jukebox by the Robot Manager. Oracle recommends not loading the drivers. If they are loaded, disable them using the Windows Device Manager.

The local media database is a binary file named `JukeboxDB`. After the file is created it is stored in the same directory as the DIVArchive Robot Manager executable files. This must first be initially populated by enabling the `RM_SCSI_AUDIT_JUKEBOX` parameter in the Robot Manager configuration file.

To successfully complete the audit, the full SCSI path of one of the DVD drives must be configured in the `RM_SCSI_AUDIT_DRIVE_ID` section in the Robot Manager configuration file. This drive is also used to audit new media as it is entered into the DVD Jukebox.

You must also enter the *Media Type* settings for the Jukebox. You obtain the settings from the `SCSI_TapeType.ini` file and entered into the `RM_SCSI_MEDIA_TYPE` parameter.

The following table identifies options in the `robotmanager.conf` configuration file:

Table 5–4   `robotmanager.conf` Options

Parameter	Parameter Type	Description	Default
RM_SCSI_AUDIT_JUKEBOX	0 (disabled) 1 (enabled)	For ASACA or PLASMON DVD Jukeboxes, set this value to 1 to audit the Jukebox and populate the `JukeboxDB` Database file. This must only be disabled at the completion of the audit.	0
RM_SCSI_AUDIT_DRIVE	ScsiP:B:T:L	This parameter contains the SCSI path of the audit drive. The syntax is `ScsiP:B:T:L` where: P = Port Number B = Bus Number T = Target ID L = Logical Unit Number	Scsi2:0:5:0
RM_SCSI_AUDIT_DRIVE_ID	0 (disabled) 1 (enabled)	This is the Drive Number that will be reported to the Configuration Utility.	0
RM_SCSI_MEDIA_TYPE	Number	This parameter sets the default *Media Type ID* (see the `SCSITape_Type.ini` file for the appropriate values).	0
RM_SCSI_UPDATE_LABELS	0 (disabled) 1 (enabled)	When the media is audited, this parameter defines whether DIVArchive will incorporate the existing DVD label into its database or create a new label. If this option is enabled, a new label is written, invalidating any existing data on the media.	1

Configuring ACSLS Attached Libraries

DIVArchive can directly interface to most Oracle StorageTek libraries using the `Robot_SCSI` driver. Some library configurations require the use of the Oracle StorageTek ACSLS library management software for the Robot Manager to control the library.

You can only install ACSLS (Automated Cartridge System Library Software) on Solaris platforms. The Solaris host and ACSLS are sold and supported by Oracle. See the Oracle ACSLS documentation at http://docs.oracle.com for detailed information.

Oracle does not support DIVArchive installations under the Solaris operating system.

Configuring LibAttach

LibAttach is an intermediate Windows driver providing connectivity to the ACSLS host. LibAttach runs as a Windows service and is typically installed on the same computer running the DIVArchive Robot Manager. The DIVArchive Robot Manager communicates to the ACSLS host using the LibAttach driver.

You must enter the following settings on the LibAttach Configurator dialog box (part of the ACSLS software):

Library server host name
Host name or IP address of the ACSLS server. If you use a host name, it must be resolvable by the DIVArchive Robot Manager host.

Firewall support
These settings are only required if a firewall is installed between the Robot Manager host and the ACSLS server. If no firewall is present leave these parameters set to `0`.

Testing the LibAttach Connectivity to ACSLS

You can verify connectivity from the Robot Manager host to the ACSLS server with the `query_server.exe` utility located in the LibAttach installation directory. When you launch the utility a Windows command prompt opens. Statistics from the library will be returned if the connection is successful.

Firewall Support

You must have a TCP or UDP port open (to allow communication) if there is a network firewall between your Robot Manager host and ACSLS server. If there is a firewall, enter the open port numbers into the *Firewall Support* settings in the LibAttach Configurator.

Early implementation of firewall support for LibAttach did not work correctly with the DIVArchive Robot Manager, even though the `query_server` utility returned a successful connection. Ensure that you have the latest release of LibAttach that incorporates the patch released to address this issue. Contact Oracle Support for additional information.

`robotmanager.conf` Common Options

The following table identifies common `robotmanager.conf` options:

Table 5–5   robotmanager.conf Common Options

Parameter	Parameter Type	Description	Default
RM_MODULE=ACSLS_ Robot.dll		Uncomment only this line	Commented
RM_PORT	TCP Port Number	The TCP Port the Robot Manager will listen on for incoming requests. This value must be unique if there are multiple Robot Managers running on a single host. The assigned port is typically TCP Port 8500 and higher.	8500

Table 5–5   (Cont.) robotmanager.conf Common Options

Parameter	Parameter Type	Description	Default
RM_ACS	Number	ACSLS configurations ignore this value because the ACS number is supplied from ACSLS.	Ignored
SERVICE_NAME	Name	This is the display name of the Robot Manager Windows service. This variable must be set if multiple Robot Managers are installed on the same server. If this variable is used, the *Service Name* will be DivaRbt-SERVICE_NAME. The *Service Name* will revert to DivaRbt if this variable is not set.	Uncommented

The following table identifies the ACSLS parameters:

Table 5–6   ACSLS Parameters

Parameter	Parameter Type	Description	Default
RM_ACSLS_SERVER	IP Address or Host Name	ACSLS ignores this parameter and it can be left blank.	
RM_ACSLS_SSI_SOCKET	TCP Port Number	ACSLS SSI socket is the UNIX domain socket used by SSI. If this value is left undefined, it defaults to TCP port 50004.	50004
RM_ACSLS_TIMEOUT	Time in milliseconds	This sets the timeout period for queries to ACSLS through LibAttach. If you leave this value set to 0, the timeout period used by the Robot Manager is 10 minutes. If you must alter this timeout period, replace 0 with your own value (in milliseconds).	0
RM_ACSLS_IE_TIMEOUT	Time in milliseconds	When an **Insert** or **Eject** tape command is issued you must open the CAP and insert or eject tapes within this timeout period. If you leave this value set to 0, the timeout period used by the Robot Manager is 10 minutes. If you must alter this timeout period, replace 0 with your own value (in milliseconds).	0

Table 5–6   (Cont.) ACSLS Parameters

Parameter	Parameter Type	Description	Default
RM_ACSLS_MAX_DISMOUNT_RETRIES	Number of retries	The maximum number of retries when the dismounted drive is still in use. If the setting is 5, the initial delay is five seconds and then doubled after each retry.	5
RM_ACSLS_DISMOUNT_FORCE	0 (disabled) 1 (enabled)	Under normal circumstances, you must unload a tape first (using an Actor) before issuing a dismount command to the library. A forced dismount instructs the library to issue the unload command to the drive directly. *This option is not recommended because this may interfere with operations on the Actors.*	0

Configuring Sony PetaServe Libraries

Control of Sony PetaServe libraries from the DIVArchive Robot Manager is directed through the Sony PSC controller over an Ethernet connection. The PSC controller parameters for the Robot Manager configuration file must match those on the PetaSite Controller.

`robotmanager.conf` Common Options

The following table identifies common `robotmanager.conf` options:

Table 5–7   `robotmanager.conf` Common options

Parameter	Parameter Type	Description	Default
RM_MODULE=SONY_Robot.dll		Uncomment only this line	Commented
RM_PORT	TCP Port Number	The TCP Port the Robot Manager will listen on for incoming requests. This value must be unique if there are multiple Robot Managers running on a single host. The assigned port is typically TCP Port 8500 and higher.	8500
RM_ACS	Number	Automated Cartridge System (ACS) controlled by the Robot Manager module.	0

The following table identifies common Sony PetaSite options:

Table 5–8   Sony PetaSite Options

Parameter	Parameter Type	Description	Default
RM_SONY_ENABLE_MEDIA_TYPE_TRIMMING	Number	*This parameter must not be modified during production. The database may need to be patched if it is changed during production.* Some tape labels contain and additional two or three characters identifying the type of media. For example, `004452L2` is an LTO2 tape and `S1000052` is a SAIT1 tape. If this parameter is set to `1`, the Sony Robot detects the tape using the label and filters out the two or three additional characters from the label.	1
RM_SONY_MEDIA_TYPE_TRIMMING_LEFT	Number	*This parameter must not be modified during production. The database may need to be patched if it is changed during production.* Depending on the label, the two characters may be on the right or on the left of the label. Set this parameter to `1` if the *Media Type* information is on the left, otherwise set it to `0`.	0
RM_SONY_PSCSERVERNAME	IP Address or Host Name	This parameter specifies the Host Name or IP Address of the Sony PetaSite controller (PSC). If you specify a Host Name, this must be defined in the operating system's `hosts` file.	
RM_SONY_PSCUSERID	Number	This specifies the *User ID* that the Robot Manager uses when it connects to the Sony PetaSite Controller.	1
RM_SONY_PSCTIMEOUT	Time in milliseconds	Command time out to the PSC in milliseconds. This is only used for mount operations.	900000

Table 5–8   (Cont.) Sony PetaSite Options

Parameter	Parameter Type	Description	Default
RM_SONY_PSCDISMOUNTRETRIES	Number of retires	The maximum number of retries when the dismounted drive is still in use. If the setting is 5, the initial delay is five seconds. The delay is then doubled after each retry.	5

Configuring ADIC Libraries with SDLC

This interface is available on both Windows and Linux platforms. Refer to Appendix D for setting up the SDLC server and client components for the DIVArchive Robot Manager interface.

robotmanager.conf Common Options

The following table identifies common robotmanager.conf options:

Table 5–9   robotmanager.conf Common options

Parameter	Parameter Type	Description	Default
RM_MODULE=ADIC_Robot.dll		Uncomment only this line	Commented
RM_PORT	TCP Port Number	The TCP Port the Robot Manager will listen on for incoming requests. This value must be unique if there are multiple Robot Managers running on a single host. The assigned port is typically TCP Port 8500 and higher.	8500
RM_ACS	Number	Automated Cartridge System (ACS) controlled by the Robot Manager module.	0

The following table identifies common ADIC parameters:

Table 5–10   ADIC Parameters

Parameter	Parameter Type	Description	Default
RM_ADIC_DAS_CLIENT	Host Name	Host Name of the computer running the ADIC DAS client.	
RM_ADIC_EJECT_AREA_NAME	Name	Symbolic name of the Cartridge Access Port.	E01
RM_ADIC_TIME_INSERT	Time in milliseconds	Number of milliseconds to wait to put away the tape after closing the CAP.	5000

Table 5–10   (Cont.) ADIC Parameters

Parameter	Parameter Type	Description	Default
RM_ADIC_MAX_DISMOUNT_ RETRIES	Number of retries	Maximum number of retries when the dismounted drive is still in use. If the setting is 5, the initial delay is five seconds. The delay is then doubled after each retry.	5

Configuring Simulated Libraries (for DIVArchive Simulators)

Simulated robots are available on Windows and Linux platforms. The settings are shown here for reference only. Refer to the *Oracle DIVArchive Simulator Operations Guide* (available to OPN partners only) for more information on installing and configuring a DIVArchive Simulator platform.

`robotmanager.conf` Common Options

The following table identifies common `robotmanager.conf` options:

Table 5–11   `robotmanager.conf` Common Options

Parameter	Parameter Type	Description	Default
RM_MODULE=SIMULATOR_Robot.dll		Uncomment only this line	Commented
RM_PORT	TCP Port Number	The TCP Port the Robot Manager will listen on for incoming requests. This value must be unique if there are multiple Robot Managers running on a single host. The assigned port is typically TCP Port 8500 and higher.	8500
RM_ACS	Number	Automated Cartridge System (ACS) controlled by the Robot Manager module.	0

The following table identifies the DIVArchive Simulator parameters:

Table 5–12   Simulator Parameters

Parameter	Parameter Type	Description	Default
RM_SIMU_BASEDIR	Directory Path	The DIVArchive simulation files base directory path. This is typically C:\Diva\Simulation.	

Table 5–12   (Cont.) Simulator Parameters

Parameter	Parameter Type	Description	Default
RM_SIMU_OPERATION_SHORT_DELAY	Time in milliseconds	This setting simulates physical delays in mount, dismount, enter, and eject operations. The recommended setting is 10000 msec.	0
RM_SIMU_OPERATION_LONG_DELAY	Time in milliseconds	You can use this setting to simulate an operation that takes more time than expected for execution. The recommended setting is 120000 msec.	0
RM_SIMU_OPERATION_LONG_DELAY_FREQUENCY	Number	This setting specifies how often a long delay should occur. The recommended setting is 50.	0
RM_SIMU_LIST_SHORT_DELAY	Time in milliseconds	This setting introduces a simulated physical delay in list operations. The recommended setting is 500.	0
RM_SIMU_LIST_LONG_DELAY	Time in milliseconds	You can use this setting to simulate a list operation that takes more time than expected for execution. The recommended setting is 60000 msec.	0
RM_SIMU_LIST_LONG_DELAY_FREQUENCY	Number	This setting specifies how often a long delay should occur in list operations. The recommended setting is 100.	0

Robot Manager Command Options

You perform DIVArchive Robot Manager control and management functions using robotmanager.exe from a command prompt. On Windows servers the executable is located in the %DIVA_HOME%\Program\Robotmanager\bin folder. On Linux servers, robotmanager.sh is located in the /home/diva/DIVA/Program/RobotManager/bin directory.

Installing and Uninstalling the Robot Manager Services in Windows

You use the following command line options to install or uninstall the DIVArchive Robot Manager from a Windows command prompt:

robotmanager -i

Installs the Robot Manager Service as set by the *SERVICE_NAME* parameter defined in `robotmanager.conf`. If this parameter is undefined, the service is installed as `DIVArchive Robot Manager - host_name`.

robotmanager -u

Removes the Robot Manager Service set by the *SERVICE_NAME* parameter in `robotmanager.conf`. If this parameter is undefined the service to be removed is `DIVArchive Robot Manager - host_name`.

These Robot Manager command options default to the `robotmanager.conf` file located in the `%DIVA_HOME%\Program\conf\robot_manager` folder to define the Service Name (if any). If you are installing multiple Robot Managers on a single host (*see Appendix A for Oracle DIVArchive options and licensing information*), additional Robot Manager configuration files must be created and specified to the service during installation to create unique instances for each Robot Manager.

You can create additional configuration files for each Robot Manager by copying and renaming the original `robotmanager.conf` file. For example, `robotmanager1.conf`, `robotmanager2.conf`, and so on. Each configuration file must contain unique *SERVICE_NAME*, *RM_PORT*, and *RM_ACS* entries.

For example, `robotmanager1.conf` might have the following parameters for a SCSI interface:

```
RM_MODULE=SCSI_Robot.dll
SERVICE_NAME=Robot1
RM_PORT=8500
RM_ACS=0
```

While `robotmanager2.conf` might have the following parameters for an ACSLS interface:

```
RM_MODULE=ACSLS_Robot.dll
SERVICE_NAME=Robot2
RM_PORT=8501
RM_ACS=1
```

You must specify the path to each Robot Manager configuration file for each instance when installing additional Robot Manager Services on the same host. You identify the path by adding the `-conf` (or `-f`) command switches when installing the service. For example, `robotmanager -i -conf ..\..\conf\robot_manager\robotmanager2.conf` installs the RObot Manager service as defined by the *SERVICE_NAME* parameter from the `robotmanager2.conf` configuration file.

If you must uninstall one or more Robot Manager Services, the configuration file path must also be specified. For example, `robotmanager -u -conf ..\..\conf\robot_manager\robotmanager2.conf` removes the Robot Manager Service as defined by the *SERVICE_NAME* parameter in `robotmanager2.conf` configuration file.

After installing the services check the Windows Services applet to confirm that the Robot Manager Services were installed correctly. To change the *SERVICE_NAME*, you must uninstall the existing service before editing the `robotmanager.conf` file. Then reinstall the service after changing the *SERVICE_NAME* parameter.

Installing and Uninstalling the Robot Manager Services in Linux

You use the following command line options to install or uninstall the DIVArchive Robot Manager from a Linux terminal.

Use the following command sequence to install the Robot Manager service:

```
cd /home/diva/DIVA/Program

./divaservice install robotmanager /home/diva/DIVA/Program/conf/robot_
manager/robotmanager.conf
```

Use the following command sequence to uninstall the Robot Manager service:

```
cd /home/diva/DIVA/Program

./divaservice uninstall robotmanager /home/diva/DIVA/Program/conf/robot_
manager/robotmanager.conf
```

See Installing the DIVArchive Services for information on the `divaservice` *command.*

Robot Manager Service Management Functions

The following command options are also available for the Robot Manager Service:

`robotmanager debug`
Starts the DIVArchive Robot Manager in console mode. Console mode displays diagnostic messages and other information from the library in the console window.

`robotmanager version`
Displays the DIVArchive Robot Manager software release information. You can also use -v instead of `version`.

robotmanager help
This displays all command line options.

Testing the Robot Manager Library Interface

After configuring the Robot Manager configuration file, launch the DIVArchive Robot Manager and confirm that the library itself can be controlled.

Library interfaces that use ACSLS, SDLC, or PSC intermediate control software must be running before launching the DIVArchive Robot Manager. ACSLS controlled libraries should also be *varied online* (for example, `vary lsm0 online`).

Starting, Stopping, and Restarting the Robot Manager

Windows DIVArchive Robot Managers start automatically with Windows. You manage (start, stop, restart, and so on) the service through the Windows Services applet.

> **Note:** If the library is offline when the service is started, the Robot Manager does not automatically reconnect after the library comes online. You must restart the service to connect.

You can also stop and then start (restart) a Robot Manager from a command window. The quotation marks in the commands must be used when specifying a service with spaces in the name. Use the following command sequence to stop and then start the service:

```
net stop "DIVArchive Robot Manager"
net start "DIVArchive RobotManager"
```

You use the following command sequence if the *SERVICE_NAME* is specified in the `robotmanager.conf` file:

```
net stop "DIVArchive Robot Manager SERVICE_NAME"
net start "DIVArchive RobotManager SERVICE_NAME"
```

Testing the Robot Manager Library Control

> **Caution:** These utilities *must not* be used in a live DIVArchive system. You *must not* send commands to a Robot Manager using either of these utilities under any circumstances when the DIVArchive Manager is running. Oracle is not responsible for any complications arising from inappropriate use of these utilities.

You can use either the Robot Manager Client (a command-line interface) or GUI to establish basic control functionality of a Robot Manager to its controlled libraries. You can use either of these utilities to send manual commands to a DIVArchive Robot Manager to initiate simple operations, for example, drive mounting, dismounting, enter or eject operations from the CAP (Cartridge Access Port). Both utilities connect to a Robot Manager through TCP/IP and can be run from a remote computer. This feature enables the Robot Manager GUI to be used from a remote computer.

If you mount a tape with either of these utilities, you must first unload the tape before it can be dismounted, unless the library supports *Forced Dismount* commands and they are enabled in the DIVArchive Robot Manager configuration file.

Robot Manager Client

This command-line client is typically located with the Robot Manager executable files in the `%DIVA_HOME%\Program\RobotManager\bin` folder.

You must specify the IP address of the Robot Manager and its TCP port when launching the client as follows:

```
RobotManagerClient {IP_Address} {TCP_Port}
```

The `IP_Address` is the IP address of the Robot Manager computer, and the `TCP_Port` is the Robot Manager listening port. You can hard-code these two parameters in the Robot Manager Client batch file if there is only a single Robot Manager requiring access.

All of the client commands are self-explanatory after you start the program.

Robot Manager Client GUI

The Robot Manager Client GUI is typically located with the Robot Manager executable files located in the `%DIVA_HOME%\Program\RobotManager\bin` folder. The GUI provides the same functionality as the command line client. You execute `RobotManagerGUI.bat` to open the GUI interface.

The GUI interface includes the following buttons and functionality:

Connect Button

Click this button to connect to the DIVArchive Robot Manager. You must enter the IP address and TCP port of the DIVArchive Robot Manager to be tested in the *Connect* prompt.

Tape List Button
Click this button to load the tape list from the library.

Reload Config. Button
Click this button to reload the configuration.

Exit Button
Click this button to exit the program.

Tape List
To manually mount a tape, select a Barcode ID and drag and drop it on to one of the drives displayed in the LSM list.

LSM List
This area lists all of the available drives and the tapes in the drive. You right-click a tape and select **Dismount** from the menu to dismount a tape.

CAP List
To manually eject a tape from the library, select a Barcode ID and drag and drop it to one of the listed CAPs.

Status Area
The Status area is at the bottom of the screen and displays status messages from the Robot Manager.

Configuring the Robot Manager at the System Level

At the system level, each instance of the DIVArchive Robot Manager must be declared to the DIVArchive Manager in the *Robot Managers* frame of the **Robots** tab in the DIVArchive Configuration Utility.

You use frame buttons to add (**+**), edit (**Edit**), or delete (**-**) a Robot Manager. The **Update** frame button refreshes the displayed Robot Manager information from the database.

Clicking the **+** button adds a Robot Manager to the configuration. The Add new row in Robot Managers dialog box is displayed. Enter the following information in the appropriate fields and then click OK to add a Robot Manager:

Name
The name of the DIVArchive Robot Manager attached to this DIVArchive system.

Address
The IP address of the host running the DIVArchive Robot Manager installation.

Port
The Robot Manager TCP port. This must match the RM_PORT parameter specified in robotmanager.conf.

Site
The DIVArchive Manager uses this parameter to determine optimal use of resources in resource allocation. Use the menu list to select the appropriate site for this Robot Manager. *Site Selection* must be enabled in the DIVArchive Manager configuration file or all sites are considered equally.

Robot Manager-ACS Association

Each DIVArchive Robot Manager is logically referred to by the DIVArchive Manager using its Automatic Cartridge System (ACS) number. This value should be unique among all DIVArchive Robot Managers. Individual libraries (or frames) are typically referred to by their Library Storage Module (LSM) number.

Use the following procedure to associate a Robot Manager with an ACS:

1. Open the Configuration Utility and connect to the database.

2. Select the **Synchronize DB** option from the **Tools** menu and acknowledge the warning message.

3. Select the individual Robot Manager to synchronize from the menu list in the Database Synchronization dialog box, or select **ALL** to synchronize all Robot Managers.

4. Only select the *Synchronize Robot Manager ACS Associations* check box. Confirm that all other check boxes are deselected.

5. Click **Go** to update the selected associations.

6. Confirm correct, successful, operation in the Status area at the bottom of the screen.

7. Enter the details for each library in the Library Data Entry dialog box when prompted, and then click **OK** to continue.

8. Click **Close** to exit the Database Synchronization dialog box.

9. Confirm the association in the *Robot Managers-ACS* frame.

Logging Robot Manager Activity

During normal operation, each DIVArchive Robot Manager logs its communications with the library and stores them in the `%DIVA_HOME%\log\robot_manager` folder. These logs are useful for troubleshooting issues. You may be asked to provide the log files when contacting Oracle Support.

The most recent log file is named `robot_manager.log` or `robot_manager_SERVICE_NAME.log` and is located in the `..\log\robot_manager` folder. Older logs are renamed with the time it was saved as its file name and moved to dated subfolders under the name of each Robot Manager.

Managing SCSI Errors

The DIVArchive SCSI Robot driver supports multiple library types with a single driver. The error codes provided from each manufacturer's library may differ. If the DIVArchive SCSI Robot Manager encounters an unexpected error from the library after a command is issued, error codes are returned from the library in the following format:

- Sense Key

- Additional Sense Code

- Additional Sense Code Qualifier

The SCSI Robot Manager first examines the `SCSI_Errors.ini` file to match these codes to a specific error message. To accommodate to each library type, library specific configuration files are provided to interpret these error codes correctly if they are not

found in `SCSI_Errors.ini`. When a match is found it is displayed in the Robot Manager command window and in the Robot Manager logs.

If you have one of the following specific library types, you must rename the corresponding file to `SCSI_Specific_Errors.ini.library_type`:

SCSI_Specific_Errors.ini.adic_s1k
For ADIC Scalar 1000 libraries

SCSI_Specific_Errors.ini.adic_s10k
For ADIC Scalar 10000 libraries

SCSI_Specific_Errors.ini.asaca
For Asaca DVD Jukeboxes

SCSI_Specific_Errors.ini.hp-esl
For Hewlett Packard libraries

SCSI_Specific_Errors.ini.ibm-3584
For IBM 3584 libraries

SCSI_Specific_Errors.ini.istora
For ADIC I-2000 libraries

SCSI_Specific_Errors.ini.storagetek
For StorageTek/SUN libraries

Configuring Media and Drive Types

After you have successfully configured the DIVArchive Robot Manager for your libraries, and the appropriate details for all DIVArchive Robot Managers entered into the **Robots** tab section of the Configuration Utility, the Tape Media, Drive Models, and the Drive Locations currently installed in each library must be entered.

The following flowchart lists the workflow of this portion of the configuration. All of the DIVArchive Robot Managers configured must be running and successfully connected to each library before commencing this portion of the configuration.
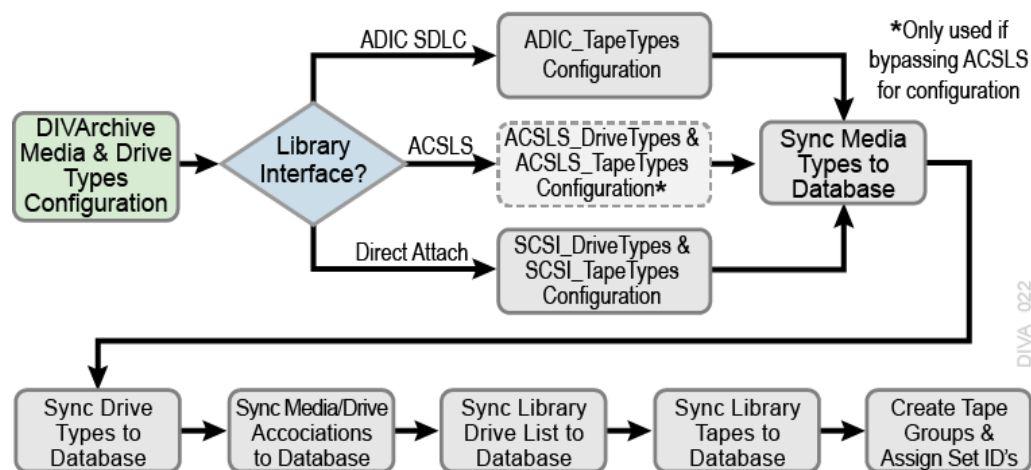
Tape Drives and their associated media types that are installed in a particular library are initially configured in the DIVArchive Database using static configuration files. The files are located in the `%DIVA_HOME%\Program\conf\robot_manager` folder. The

DIVArchive Robot Manager selects the appropriate files according to the `RM_MODULE` setting configured in `robotmanager.conf`.

The following list identifies the configuration file names and use:

`scsi_drive_types.ini` and `scsi_tape_types.ini`
Used for direct attached SCSI libraries. These files are only considered if the `.ini` extension is removed.

`acsls_drive_types.ini` and `acsls_tape_types.ini`
Used for libraries managed by ACSLS. Normally, tape and drive types are derived from ACSLS during library synchronization with the database. However, you can use these files to override the values returned from ACSLS. These files are only considered if the `.ini` extension is removed.

`adic_media_types.ini`
Used with ADIC libraries controlled by SDLC. Drive Types for this library are directly returned from the SDLC server. These files are only considered if the `.ini` extension is removed.

When a hardware audit is initiated on the specific library by the Configuration Utility (through the DIVArchive Robot Managers, either directly or through intermediate library management software), hexadecimal codes are returned to identify the model and order of the tape drives currently installed, and the media types present in the library.

These library hardware codes are mapped to drive and media IDs within the DIVArchive Database using the *Tape Types* and *Drive Types* configuration files.

It is only necessary to modify these files when Drive Types or Media Types are added to the library.

SCSI_drive_types and ACSLS_drive_types

You can edit these files using any plain text editor (for example, Notepad or Notepad++). No modification of these files is required other than to remove comment fields for the appropriate library and drive types for your installation.

Remove the # at the beginning of the line in the appropriate library section for the drives to be recognized in a **Synchronize Drive Types List** in the Configuration Utility. You must leave drive types in libraries not installed commented out.

The *Compatible Drive Types* column cross-references the *Tape Type ID* in `SCSI_Tape_Types` (or `ACSLS_Tape_Types` if used). These values are examined in the **Synchronize Media/Drive Compatibility List** procedure in the Configuration Utility.

SCSI_tape_types and ACSLS_tape_types

You can edit these files using any plain text editor (for example, Notepad or Notepad++). No modification of these files is required other than to remove comment fields for the tape types for your specific library.

Remove the # at the beginning of the line in the appropriate library section for the tapes or DVDs to be recognized in a **Synchronize Media Types List** in the Configuration Utility. You must leave tape types (or DVDs) in libraries not installed commented out.

The *Compatible Drive Types* column cross-references the *Drive Type ID* in `SCSI_Drive_Types` (or `ACSLS_Drive_Types` if used). These values are examined in the **Synchronize Media/Drive Compatibility List** procedure in the Configuration Utility.

ADIC_media_types

You can edit these files using any plain text editor (for example, Notepad or Notepad++). No modification of these files is required other than to remove comment fields for the tape types for your specific library.

Remove the # at the beginning of the line in the appropriate library section for the tapes to be recognized in a **Synchronize Media Types List** in the Configuration Utility. You must leave tape types in libraries not installed commented out.

The *Compatible Drive Types* column cross-references the *Drive Type ID* returned from the SDLC controller. These values are examined in the **Synchronize Media/Drive Compatibility List** procedure in the Configuration Utility.

Defining Tape Capacity and Block Sizes

The values in the following two tables must be used when entering adding a Drive Type or Media Type in the DIVArchive Database. The values have been tuned by Oracle to avoid tape spanning, and therefore may be lower than the theoretical capacity.

The following table identifies tape capacities to use when entering a Drive Type or Media Type in the database:

Table 5–13   Tape Capacity Definitions

Media Type	Drive Type	Capacity
9840	STK 9840A	19 531 008
	STK 9840B	19 531 008
	STK 9840C	39 062 272
9940	STK T9940A	58 593 536
	STK T9940B	195 312 384
T10000T1	STK T10000A	488 281 008
	STK T10000B	976 562 176
T10000TS	STK T10000A	117 187 072
	STK T10000B	234 374 656
	STK T10000C	5 243 000 000
	STK T10000D	7 812 500 480
	STK T10000D (maximum capacity enabled)	8 300 781 056
DTF-2	GY-8240	195 312 448
SAIT1	S-AIT 1	488 281 088
SAIT2	S-AIT 2	781 249 536
AIT3	AIT 3	97 656 192
LTO-5	IBM or HP	1 464 843 264
LTO-6	IBM or HP	2 441 405 952
LTO-7	IBM	5 859 374 592
LTO-100G	IBM, HP, Seagate LTO Ultrium 1	97 656 192
LTO-200G	IBM LTO Ultrium 2	195 312 128
LTO-400G	IBM or HP LTO Ultrium 3	390 624 768

Table 5–13   (Cont.) Tape Capacity Definitions

Media Type	Drive Type	Capacity
LTO-800G	IBM or HP Ultrium 4	781 249 536
DLT-IV	Quantum DLT7000	34 179 648

The following table identifies tape block sizes to use when entering a Drive Type or Media Type in the database:

Table 5–14   Tape Block size Definitions

Manufacturer	Tape Drive Type	Block Size in Bytes
HP	LTO Ultrium 1	65536
	LTO Ultrium 2	524288
	LTO Ultrium 3	524288
	LTO Ultrium 4	524288
IBM	LTO Ultrium 1	65536
	LTO Ultrium 2	524288
	LTO Ultrium 3	524288
	LTO Ultrium 4	524288
	LTO-5	524288
	LTO-6	524288
	LTO-7	524288
Oracle StorageTek	T9840A	262144
	T9840B	262144
	T9840C	262144
	T9940A	262144
	T9940B	262144
	T10000A	524288
	T10000B	524288
	T10000C	524288
	T10000D	524288
Quantum	DLT 7000	65536
Seagate	LTO Ultrium 1	65536
Sony	GY-8240 (DTF-2)	65536
	AIT-3	65536
	S-AIT 1	524288
	S-AIT 2	262144

Synchronizing Media Types with the Database

You must import the values that have been uncommented in the `Tape_Types` configuration files into the DIVArchive Database. Each DIVArchive Robot Manager to be queried must be online to complete this procedure successfully.

Use the following procedure to import and synchronize the values from the `Tape_Types` files in the database:

> **Caution:** Only perform this operation if you are adding Media Types to the library.

1. Open the Configuration Utility and connect to the database.

2. Select the **Synchronize DB** option from the **Tools** menu and acknowledge the warning message.

3. Select the individual Robot Manager to synchronize from the menu list in the Database Synchronization dialog box, or select **ALL** to synchronize all Robot Managers.

4. Only select the *Synchronize media types list* check box. Confirm that all other check boxes are deselected.

5. Click **Go** to update the selected associations.

   The Configuration Utility will connect to the DIVArchive Robot Manager. The Robot Manager parses the SCSI_Tape_Types (or ACSLS_Tape_Types if used) configuration file.

6. If a Tape Type is not currently in the database, you will be prompted to enter it. Click **No** for any Tape Types not currently in use.

   > **Note:** If you report cleaning tapes in the following two steps, you must enter a *Tape Size* and *Block Size* of 1 KB for each cleaning tape added so they do not interfere with the total available size computation of all tapes in the Control GUI.

7. Enter the *Total Size* for this Media Type and click **OK**.

8. Enter the *Block Size* for this Media Type. Ensure you enter the *Block Size* correctly before clicking **OK** because you cannot change it later.

9. Click **Close** to exit the Database Synchronization dialog box.

10. Confirm the Tape Type has been correctly entered in the *Tape Properties* frame of the Configuration Utility **Tapes** tab.

Synchronizing Drive Types with the Database

You must also import the uncommented values in the Drive_Types configuration files into the DIVArchive Database. Each DIVArchive Robot Manager to be queried must be online to complete this procedure successfully.

Use the following procedure to import and synchronize the values from the Drive_Types files in the database:

> **Caution:** Only perform this operation if you are adding Drive Types to the library.

1. Open the Configuration Utility and connect to the database.

2. Select the **Synchronize DB** option from the **Tools** menu and acknowledge the warning message.

3. Select the individual Robot Manager to synchronize from the menu list in the Database Synchronization dialog box, or select **ALL** to synchronize all Robot Managers.

4. Only select the *Synchronize drive types list* check box. Confirm that all other check boxes are deselected.

5. Click **Go** to update the selected associations.

   The Configuration Utility will connect to the DIVArchive Robot Manager. The Robot Manager parses the `SCSI_Drive_Types` (or `ACSLS_Drive_Types` if used) configuration file.

6. If a Drive Type is not currently in the database, you will be prompted to enter it. Click **No** for any Drive Types not currently in use.

7. Enter the *Block Size* for this Drive Type. Ensure you enter the *Block Size* correctly before clicking **OK** because you cannot change it later.

8. Confirm there are no errors in the status window. If errors appear, recheck the `Tape_Types` and `Drive_Types` definition files.

9. Click **Close** to exit the Database Synchronization dialog box.

10. Confirm the Drive Type has been correctly entered in the *Drive Properties* frame of the Configuration Utility **Drives** tab.

Synchronizing Media and Drive Compatibility with the Database

This step cross-references the compatibility entries in the `Tape_Types` and `Drive_Types` definition files.

For libraries controlled by ACSLS Media and Drive Type, information is normally retrieved directly from ACSLS,. Therefore, an ACSLS software upgrade or a library firmware update may require the Media and Drive Type settings to be resynchronized.

Use the following procedure to synchronize the media and drive compatibility in the database:

> **Caution:** Only perform this procedure is you are adding a Media or Drive Type, or updates are made to the Tape or Drive Types definition files in a DIVArchive software update.

1. Open the Configuration Utility and connect to the database.

2. Select the **Synchronize DB** option from the **Tools** menu and acknowledge the warning message.

3. Select the individual Robot Manager to synchronize from the menu list in the Database Synchronization dialog box, or select **ALL** to synchronize all Robot Managers.

4. Only select the *Synchronize media/drive compatibility list* check box. Confirm that all other check boxes are deselected.

5. Click **Go** to update the selected associations.

   The Configuration Utility will connect to the DIVArchive Robot Manager. The Robot Manager parses the `SCSI_Tape_Types` (or `ACSLS_Tapes_Types` if used) configuration file.

6. Confirm there are no errors in the status window. If errors appear, recheck the `Tape_Types` and `Drive_Types` definition files.

7. Click **Close** to exit the Database Synchronization dialog box.

8. Confirm the Media Type and Drive Type associations have been correctly entered in the *Media Compatibility* frame of the Configuration Utility **Robots** tab.

Synchronizing the Library Drive List with the Database

If you add Drive Types or additional drives to a DIVArchive Managed Library, you must declare them in the DIVArchive Database. Drives that are added are initially set **Offline**, and therefore disabled. Before they can be used, you must set them **Online** and notify the Manager (if running). During DIVArchive operations, the Manager may automatically set a drive **Offline** if it encounters a problem with it.

When the *Used* field in the *Drives* frame is set to **N**, DIVArchive ignores the drive and it is not displayed in the Control GUI **Drives** tab. If you subsequently set a drive to **Y**, DIVArchive will not use it until you notify the Manager. This field restricts using drives in libraries that are shared with other backup or archive applications.

The Operations field in the Drives frame defines which operations are permitted on each drive. Operations can be one of the following:

R
The drive is dedicated to only **Repack** operations.

S
The drive will perform all standard operations only. That is, all operations except **Repack**.

A
The drive can perform all operations *including* **Repack**.

N
The drive will not be used for any operations. However, it can be enabled later without a Manager restart.

Use the following procedure to add the drives to the database:

1. Open the Configuration Utility and connect to the database.

2. Select the **Synchronize DB** option from the **Tools** menu and acknowledge the warning message.

3. Select the individual Robot Manager to synchronize from the menu list in the Database Synchronization dialog box, or select **ALL** to synchronize all Robot Managers.

4. Only select the *Synchronize drive list* check box. Confirm that all other check boxes are deselected.

5. Click **Go** to update the selected associations.

   The Configuration Utility will connect to the DIVArchive Robot Manager. The Robot Manager obtains the current drive list and drive location for each drive from the library.

6. Confirm there are no errors in the status window. If the drives reported from the library do not match those declared in the *Drive Properties* frame, an error is displayed and no drives are entered into the database.

7. Click **Close** to exit the Database Synchronization dialog box.

8. Confirm the drives have been correctly entered in the *Drives* frame of the Configuration Utility **Drives** tab.

Manually Identifying Drive Serial Numbers

When using a tape library with DIVArchive there are two logical connections to each drive in that library. The first is the *Robotics Control* (managed by the DIVArchive Robot Manager) for mounting and dismounting the tapes from specific drives, and the *Data Interface* to the drive from the Actors.

Tape libraries identify their drives by the *Drive ID* (typically 0, 1, 2, and so on). DIVArchive needs to know the corresponding data path to that drive from each Actor when the Robot Manager instructs the library to mount a tape to a specific Drive ID. If the Actor-Library mapping is incorrect, DIVArchive attempts to read or write to the incorrect drive, resulting in possible data loss or corruption.

The host computer operating system presents each drive to applications using their SCSI ID. The SCSI ID for a drive can vary as hardware is added or removed. This is particularly true when shared among multiple hosts in a SAN based environment. This configuration requires statically configured SCSI IDs using persistent bindings. This configuration dramatically complicates drive replacement.

To simplify configuration and streamline future drive replacements, the data path mapping to each drive (for its physical location in the library) is achieved by using its unique serial number rather than its SCSI ID. When a DIVArchive Actor is launched it interrogates each drive's serial number and compares it to the values in the database. Then the Actor establishes the correct data path to the drive, irrespective of its SCSI ID.

Each drive's serial number is automatically identified by library synchronization with the database during initial installation or drive replacement. Some cases may require you to manually determine the serial number and enter it into, or verify it against, the database.

You can manually identify the drive serial number either using the library's front panel display, or using the Scandrive Utility and the Robot Manager Client or GUI.

The latter method involves mounting a tape into a specific drive number in the tape library, establishing which drive the Actor is reporting that has that tape mounted, and then recording its serial number and entering, or verifying, it with the corresponding library Drive ID in the database. You must only complete this process one time for each drive in the library.

> **Caution:** The Robot Manager Client GUI utility issues direct commands to the Robot Managers and will interfere with DIVArchive operations. It interacts directly with both the Robot Managers and the Tape Drives in the libraries. *You must not use it while the DIVArchive Manager is running.*

You can use the Robot Manager Client GUI utility to send manual mount commands to a DIVArchive Robot Manager. See the Robot Manager Client GUI section for information.

The serial number of each drive can be discovered by using the `scandrive.exe` utility located in the `%DIVA_HOME%\Program\Actor\bin` folder. This utility automatically reports all SCSI devices installed in the host computer, and their corresponding port,

bus, target and logical unit numbers. For tape devices, the utility also indicates the drive's firmware, serial number, and whether a tape is loaded into the drive.
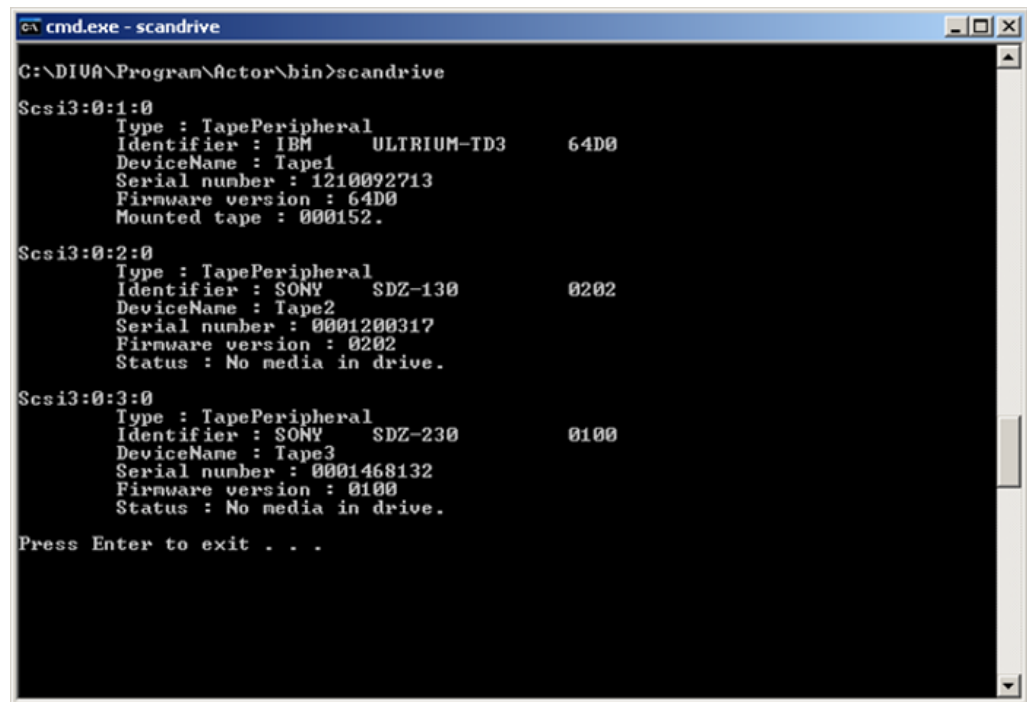
After a tape is mounted in a drive (using the Robot Manager Client GUI), run the `scandrive.exe` utility on an Actor host (that will use the selected drive) to determine which drive has the tape mounted and its corresponding serial number.

See Determining the SCSI Library Connection for information on using the Scandrive utility.

In the following figure the *Type* section refers to that peripheral's class (HDD, CDROM and so on). Each tape drive will be reported as a **TapePeripheral**, and the *Identifier* for each corresponding device should match the model number of the drive itself (for example, IBM Ultrium TD2).

Confirm the tape barcode is the correct one loaded through the Robot Manager Client GUI. You must then enter the serial number for the appropriate drive by highlighting it in the *Drives* section of the Configuration Utility, and then selecting **Edit**. Repeat the process by mounting a tape into the next library drive.

Remember to dismount the tape after determining the drive's serial number.

```
cmd.exe - scandrive                                              _|□|×|

C:\DIVA\Program\Actor\bin>scandrive

Scsi3:0:1:0
        Type : TapePeripheral
        Identifier : IBM       ULTRIUM-TD3      64D0
        DeviceName : Tape1
        Serial number : 1210092713
        Firmware version : 64D0
        Mounted tape : 000152.

Scsi3:0:2:0
        Type : TapePeripheral
        Identifier : SONY      SDZ-130          0202
        DeviceName : Tape2
        Serial number : 0001200317
        Firmware version : 0202
        Status : No media in drive.

Scsi3:0:3:0
        Type : TapePeripheral
        Identifier : SONY      SDZ-230          0100
        DeviceName : Tape3
        Serial number : 0001468132
        Firmware version : 0100
        Status : No media in drive.

Press Enter to exit . . .
```

Synchronizing the Library Tapes with the Database

Each tape inserted into a library is initially identified by its barcode label. DIVArchive keeps track of tapes currently in the library and that have been externalized in its database.

The labels and status (whether internalized or externalized) are updated in the database by **Insert Tape** or **Eject Tape** commands issued to DIVArchive. The database can become out of synch with a library's contents when tapes are added or removed directly in the library rather than through DIVArchive.

Use the following procedure to re-synchronize the tape list in the database with the library contents:

> **Tip:** This procedure is a quick way to populate the database with tapes from the library when tapes are initially loaded.

1. Open the Configuration Utility and connect to the database.

2. Select the **Synchronize DB** option from the **Tools** menu and acknowledge the warning message.

3. Select the individual Robot Manager to synchronize from the menu list in the Database Synchronization dialog box, or select **ALL** to synchronize all Robot Managers.

4. Only select the *Synchronize tape list (can be very long)* check box. Confirm that all other check boxes are deselected.

5. Click **Go** to update the selected associations.

   The Configuration Utility will connect to the DIVArchive Robot Manager. The Robot Manager obtains the current tape list from the library.

   Tapes in the library are compared to the tape tables in the DIVArchive Database. New tapes are inserted into the table and existing tapes have their status updated (internalized or externalized).

6. If a Tape Type is reported that does not match the types configured in the *Tape Properties* frame, an error is reported and no update of the database occurs. This type of error can also occur if a library cannot correctly read a tape's barcode label. You must carefully check the Robot Manager logs in this case.

7. Click **Close** to exit the Database Synchronization dialog box.

8. New tapes discovered during the audit are added to the *Unused Tape Sets* frame in the **Sets, Groups, & Media Mapping** tab of the Configuration Utility, and assigned *Set ID 1*. Tapes currently tracked by DIVArchive that are missing from the audit will have their status updated to *externalized*. You can examine the status of all tapes in the DIVArchive Control GUI.

Creating Tape Groups

You use the **Sets, Groups, & Media Mapping** tab of the Configuration Utility to define Tape Groups within the archive. Groups segment material within the tape library, or associate content with a particular Media Type. The default group is present in all installations and cannot be removed. However, you can specify your own Group Names and not use the default group. Generally, the Group Name is descriptive of the function or content that is being stored.

A group is associated with a *Set ID* defining the pool of tapes it can draw upon to store DIVArchive objects. When DIVArchive writes an object to a tape from the pool, the tape is assigned to a group. It is released from the group when all objects have been deleted or the tape has been repacked.

The group concept in combination with the Set ID enables optimal use of tape resources. Some tape drives and media are extremely fast but typically have less storage than their larger capacity (and slower) counterparts. Content that is small, or required very quickly, should be archived to this group and should use the faster drives.

For example, the 9840C tape drive is small in capacity, but it provides extremely fast access times (approximately fifteen seconds from mount to data retrieval), and is better suited to storing large numbers of relatively small data files. This is particularly true related to tape repacking.

For example, if the *Commercials Group* is allocated *Set ID 3* and all 9840C tapes are assigned to that set. Short form commercial material written to tape will exclusively use the 9840C media. Longer (and larger) material, such as movies and interstitial programs are better suited to the larger capacity tape sets.

See Sets, Groups & Media Mapping Tab Frames for information displayed on the Groups frame.

Creating Tape Sets

When a new tape is entered into a library, or DIVArchive clears a tape of its objects (whether all objects on that tape have been deleted, migrated to another tape, or moved to another tape after a tape repack), the tape is released back to the *Unused Tapes Sets* pool.

New tapes are automatically assigned a *Set ID1*, which is the default in all DIVArchive installations. Other *Set ID* numbers are typically used to distinguish between different types of media, but could be used to create restricted pools of tapes for particular applications. If this is the case in your installation, the *Set ID* must be updated for these tapes after they are inserted into the library.

See Sets, Groups & Media Mapping Tab Frames for information displayed on the Unused Tape Sets frame.

Remapping Media

You can put transformation rules in place for the specified groups on Archive requests on the *Media Mapping* frame in the Sets, Groups & Media Mapping Tab. The remapped destination media can be either a disk array, tape group, or a storage plan. This is not typically used during initial installation, but rather at a later time in the object's life cycle.

Transformation rules allow transparent redirection of objects from one media type to another without needing to alter the archive initiator. Some examples are migration of an existing group to a new drive or tape generation, or migration from tape to disk.

> **Note:** You must use a migration job to change a tape format from Legacy to AXF. Repacking a tape will not change the tape format. Repacking of existing Legacy format objects retains the format of the tape even if the tape group format was updated in the configuration from Legacy to AXF.

The following events appear in the request details when an object's media is remapped to another media, a storage plan, or both:

- *Media Name Translation* has changed the *Destination Media* to **media**.

- *Media Name Translation* has changed the *Destination Media* to **storageplan**.

- *Media Name Translation* has changed the *Destination Media* to **media & storageplan**.

6

Configuring DIVArchive Actors

This chapter describes DIVArchive Actor configuration and operations, and includes the following information:

- Configuration Overview
- Configuring the Local Actor (`actor.conf`)
- Configuring Oracle DIVArchive Partial File Restore
- Defining and Declaring Actors
- Defining Actor to Disk Connections
- Actor to Disk Interfaces and Mount Points
  - Local Interface
  - Remote Interface
  - Oracle Storage Cloud Interface
  - BML Interface
  - FTP Interface
  - MetaSAN Interface
  - Simulation Interface
- Configuring Actor to Drive Connections
- Logging Actor Activity
- Installing and Uninstalling Actor Services in Windows
- Installing and Uninstalling Actor Services in Linux
- Actor Service Management Functions
- Launching the Actors

Configuration Overview

The DIVArchive Actor runs on both Windows and Linux platforms. Windows Actors no longer start automatically with Windows. The DIVArchive Actor runs as a standalone server application. The DIVArchive Manager connects to each Actor as a client application.

The Actor is installed in the `%DIVA_HOME%\Program\Actor\bin\` folder in Windows, and in the `/home/diva/DIVA/Program/actor/bin/` directory in Linux. The Actor's configuration files are located separately in the `%DIVA_HOME%\Program\conf\Actor\`

folder in Windows, and in the `/home/diva/DIVA/Program/conf/actor/` directory in Linux. At the system level, the location and capabilities of each DIVArchive Actor are defined in the Configuration Utility.

The Actor configuration parameters are located the Configuration Utility, except for the *Service Name* and *Port*. These settings are located under **Actor Advanced** and **Partial Restore Settings** tabs of the *Actor* frame of the **System** tab. *Some settings are only available In Engineering Mode.*

You must notify the actors of any changes to the configuration by clicking on **Notification**, **Notify Actors** while connected to the Manager. The actors must be running and connected to the Manager to receive the notifications.

The following figure is the workflow for installing a DIVArchive Actor:
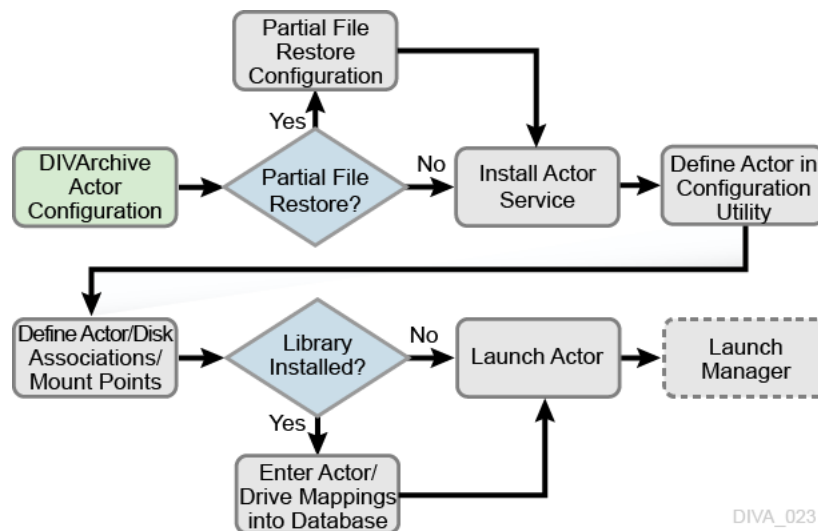
Configuring the Local Actor (`actor.conf`)

The Actor configuration file contains the *Service Name* and *Port* parameters. Remove the `.ini` extension from the `actor.conf.ini` file and edit the file with a plain text editor (for example, Notepad or Notepad++) to insert the *Service Name* and *Port* number as described in the following table.

Table 6–1   `actor.conf` Configuration Parameters

Parameter	Parameter Type	Description	Default
`DIVAACTOR_ PORT`	TCP Port Number	The TCP Port Number for the Actor to listen on for incoming requests. If running more than one Actor on the host, the TCP Port Number must be unique for each Actor.	9900
`SERVICE_NAME`	Name	The `DIVAACTOR_SERVICE_NAME` parameter specifies the name of the Actor and the service during installation. This is required if you install two or more Actors on a single Windows host computer because both cannot have the same Actor Service Name. If this parameter is not defined or commented out, the Service Name defaults to the Host Name of the Actor computer and will be `DivaAct Host_Name`.	

Configuring Oracle DIVArchive Partial File Restore

The Partial File Restore parameters are located on the **Partial Restore Settings** tab in the Configuration Utility *Actor* frame. These options provide additional parameters to the Actor for specific partial file restore formats.

To edit the parameters, double-click the Actor Name in the **Partial Restore Settings** tab to open the Edit Partial Restore Settings dialog box. The Partial File Restore options are defined on the **Partial Restore Settings** tab of the dialog box.

The following table describes the Partial File Restore parameters available on the Edit Partial Restore Settings Entry dialog box:

Table 6–2   Oracle Partial File Restore Parameters

Parameter	Value or Type	Description	Default
Name	String	This is the name of the Actor associated with these Partial File Restore options. This value is automatically filled in from the Actor settings. If you modify the name here, or in the Actor settings screen, it will be modified in both places.	
QT Ignore Start Timecode	**N** (disabled) **Y** (enabled)	If this setting is enabled, Partial File Restore will ignore the SOM value of the original clip and process TCIN and TCOUT as if it starts from 00:00:00:00.	**N**
QT Omneon First Frame Handling	**IGNORE** **RESET** **UPDATE**	This setting identifies how the Actor will handle the first frame of a QuickTime clip: ■ **IGNORE**: Partial Files Restore will ignore this field. The value found in the original clip will remain unchanged in the restored clip. ■ **RESET**: Partial File Restore will reset the value of this field to zero. ■ **UPDATE**: Partial File Restore will increment this value using the frame count from which the partially restored file begins.	**RESET**
AVI Ignore Start Timecode	**N** (disabled) **Y** (enabled)	If this setting is enabled, Partial File Restore will ignore the SOM value of the original clip and process TCIN and TCOUT as if it starts from 00:00:00:00.	**N**
EVS MXF Ignore Start Timecode	**N** (disabled) **Y** (enabled)	If this setting is enabled, Partial File Restore will ignore the SOM value of the original clip and process TCIN and TCOUT as if it starts from 00:00:00:00.	**N**

Table 6–2 (Cont.) Oracle Partial File Restore Parameters

Parameter	Value or Type	Description	Default
GXF Timecode Reference	Integer	This setting specifies how the time code SOM reference is to be derived for a GXF Partial File Restore request. The options are defined by the following values: ■ The objects start time codes are ignored. TCIN and TCOUT must be relative to 00:00:00:00. ■ SOM is derived from the first field number of the MAP packet (default). ■ SOM is derived from the time code at **Mark In** from the UMF packet.	1
GXF Progressive Timecode Translation	**N** (disabled) **Y** (enabled)	Partial File Restore is expecting TCIN and TCOUT to be in conformance with the frame rate of the archived clip by default. For example, if the frame rate of the clip is 29.97fps NTSC (or 25fps for PAL), the frame count of TCIN and TCOUT can be comprised between 0 and 29 (25 if it is PAL). HD formats have progressive frame rates (23.976, 24, 29.97, 30, 59.94, 60). For automations, the actual frame rate of the clip can be unknown. When this parameter is set to **Y** (enabled), DIVArchive considers that TCIN and TCOUT are PAL or NTSC timecodes and translates these timecodes according to the actual frame rate of the archived clip.	**N**
LXF Ignore Start Timecode	**N** (disabled) **Y** (enabled)	If this setting is enabled, Partial File Restore will ignore the SOM value of the original clip and process TCIN and TCOUT as if it starts from 00:00:00:00.	**N**
MXF Partial Restore Dictionary File	Path and File Name	This parameter must point to the name and location of the MXF dictionary file. The dictionary is normally distributed with the DIVArchive Actor installation in the `%DIVA_HOME%\Program\Actor\bin` folder. The default dictionary file name is `mxf_file.bin`. Set this parameter to `%DIVA_HOME%\Program\Actor\bin\mxf_file.bin`. Where `%DIVA_HOME%` is the root path of your DIVArchive installation for the Actor (typically `C:\Diva` in Windows and `home/diva/DIVA` in Linux).	

Table 6–2 (Cont.) Oracle Partial File Restore Parameters

Parameter	Value or Type	Description	Default
MXF Timecode From Source Package	**N** (disabled) **Y** (enabled)	If you set this parameter **Y** (enabled), the time code track used to locate the in and out points will be the one from the source package. Otherwise, timecode will be sourced from the Material Package.	**N**
MXF Timecode Value To Switch Package	**-1** (no switch) **0** (switch)	If the SOM value found in the MXF package specified by the parameter *MXF Timecode From Source Package* is equal to this value, the Actor will automatically look for the SOM in the other MXF Package. The default value of **-1** avoids switching from one package to the other.	**-1**
MXF Enforce Closed Header	**N** (disabled) **Y** (enabled)	If this parameter is set to **Y** (enabled) the extraction will fail if the metadata in the header is not closed. If set to **N** (disabled), the Actor will attempt to find closed metadata in the footer partition.	**Y**
MXF Run In Processor	File Name	If this parameter is defined it must contain the name of the RunInProcessor.dll. In this case, the run-in processor will be used to read and create run-ins. For example: RUN_IN_ PROCESSOR=RunInProcessor.dll.	
MXF Ignore Start Timecode	**N** (disabled) **Y** (enabled)	If this parameter is set to **Y** (enabled), MXF Partial File Restore will ignore all start time code values of the original clip and TCIN and TCOUT (SOM and EOM) is processed as if the original clip starts at 00:00:00:00. This option overrides the *MXF TIMECODE FROM SOURCE PACKAGE* parameter.	**N**
MXF Use Omneon Dark Meta	**N** (disabled) **Y** (enabled)	Certain Omneon MXF clips have their start time code located in a *Dark Metadata Set*. By default the MXF Partial File Restore does not pay attention to this field. Set this parameter to **Y** if you want the MXF Partial File Restore to manage this field.	**N**
MXF Serialize Depth First	**N** (disabled) **Y** (enabled)	If this parameter is set to **Y** (enabled) the MXF Partial File Restore serializes the Metadata Sets of the partially restored clip using a depth-first approach. This option is recommended when the destination is a QUANTEL ISA gateway. If it is set to **N** (disabled), the MXF Partial File Restore serializes the Metadata Sets with no ordering.	**N**

Table 6–2   (Cont.) Oracle Partial File Restore Parameters

Parameter	Value or Type	Description	Default
MXF Generate Random Index Pack	**N** (disabled) **Y** (enabled)	Random Index Pack (RIP) is an optional small structure located after an MXF file that contains file offset information for each partition in the file (when present). You can set this parameter to **N** (disabled), for incompatible servers (for example, SONY XDCAM).	**Y**
MXF Number of Frames Per Body Partition	Integer between **50** and **250**.	This parameter defines the number of frames per partition in the output file. Only values between 50 and 250 are valid. If a value greater than 250 is entered, the MXF Partial File Restore will use 250. If the entered value is less than 50, it will use 50. *This parameter is rounded automatically by the Actor to align body partitions on GOP boundaries.*	**250**
MXF Update TC Track Origin	**N** (disabled) **Y** (enabled)	When the video essence is MPEG2 LGOP, Partial File Restore will use the origin field of each track to be frame accurate. The origin specifies GOP precharge frames. Your video server may use a different implementation or interpretation of this field. If this parameter is set to **Y** (enabled), the *Origin* field is modified in *all* tracks. If this parameter is set to **N** (disabled), the *Origin* field is modified in all tracks *except* the timecode track.	**N**
MXF Tolerance on TCOUT	Integer between **0** and **250**.	This parameter can be set to indicate a tolerance on the TCOUT supplied to a Partial File Restore request. This tolerance value is 0 by default, but it you can set it to a specific number of frames. If the supplied TCOUT is beyond the end of the clip, but not too far out (within the tolerance), DIVArchive will perform the Partial File Restore until the end of the clip instead of reporting and invalid TCOUT.	**0**
MXF Duration From Footer	**N** (disabled) **Y** (enabled)	When the duration of the input clip is -1 in the header partition, the MXF Partial File Restore loads the footer partition in to obtain the correct value. Some older clips may not have a correct RIP after the file, and the footer partition may not be accessible. If you set this value to **N** (disabled), the MXF Partial File Restore does not load the footer partition and performs a blind Partial File Restore, if TCIN and TCOUT are valid.	**Y**

Table 6–2   (Cont.) Oracle Partial File Restore Parameters

Parameter	Value or Type	Description	Default
MXF Maximum Queue Size	Integer between **0** and **200**.	The maximum size (in MB) that the extractor can queue before producing an error (to avoid running out of memory).	**200**
Seachange Ignore Start Timecode	**N** (disabled) **Y** (enabled)	If you set this parameter to **Y** (enabled), SeaChange Partial File Restore ignores the start time code value of the original clip and processes TCIN and TCOUT as if it starts from 00:00:00:00. The configuration of the MXF parser is also required for MXF. However, because this is a SeaChange clip, it ignores the *MXF Ignore Start Timecode* in this workflow.	**N**
MPEG2 Transport Stream Ignore Start Timecode	**N** (disabled) **Y** (enabled)	If you set this parameter to **Y** (enabled), the MPEG2 transport stream Partial File Restore ignores the start time code value of the original clip, and processes TCIN and TCOUT as if it starts from 00:00:00:00.	**N**
MPEG2 Program Stream Ignore Start Timecode	**N** (disabled) **Y** (enabled)	If you se this parameter to **Y** (enabled), MPEG2 transport stream Partial File Restore ignores the start timecode value of the original clip and processes TCIN and TCOUT as if it starts from 00:00:00:00.	**N**

Defining and Declaring Actors

Each DIVArchive Actor must be declared in the DIVArchive Database. You declare the Actors in the *Actors* frame in the Configuration Utility's **System** tab. The *Actors* frame has three tabs:

Actor Settings
This tab includes general Actor definition settings such as Actor name, IP address, port, production system, and so on.

Actor Advanced Settings
This tab includes advanced settings such as read and write block sizes, tape unit timeout, Quantel, QuickTime and FTP settings.

Partial Restore Settings
This tab includes Partial File Restore settings previously in the Partial File Restore configuration file.

Actor and Partial File Restore settings are configured and edited on the Actor Settings Entry screen. Click **+** on the top right of the *Actor Settings* frame to create and configure an Actor, or double-click the actor you want to edit to access the settings screen.

The following list describes the maximum operations parameters on the Actor Settings Entry screen:

Name

This is the name of the Actor associated with the *Partial File Restore* options. This value is automatically filled in from the Actor settings. If you modify the name here, or in the Actor Settings Screen, it will be modified in both places.

IP Address

This is the IP address of the Actor.

Port

This is the port number the Actor listens on for commands.

Prod. System

This parameter identifies the production system where the Actor is in use.

Site

This parameter identifies the physical location of the production system.

Max Drive Operations

This is the maximum number of simultaneous requests to and from drives that this Actor can perform. You can use this parameter to distribute requests and bandwidth among all Actors.

Max Server Operations

This is the maximum number of simultaneous requests to and from servers from the Sources and Destinations configuration that this Actor can perform. You can use this parameter to distribute requests and bandwidth among all Actors.

Max Disk Operations

This is the maximum number of simultaneous transfers to and from disks (both read and write) that this Actor can perform. You can use this parameter to distribute requests and bandwidth among all Actors.

Verify Tape

This parameter defines whether tapes are verified.

Direct Restore

This parameter defines whether this Actor can be used for direct restores to a Source or Destination.

Cache Restore

The Actor is permitted to perform cache restores to a Source or Destination. You must disable this option if this Actor has no local cache storage for the temporary storage of the DIVArchive object during a transfer.

Copy To Group

This parameter defines whether this Actor can be used for Copy To Group requests. You can use this option to isolate specific Actors involved in critical operations from mass Copy To Group requests, such as those from the DIVArchive SPM option.

Associative Copy

This parameter defines whether this Actor can be used for Associative Copy requests.

Repack

This parameter defines whether this Actor can be used for tape repack requests. You must set this to **N** if the Actor has no local cache for temporary storage during the repack operation. Because tape repacking is a lengthy operation, you can also use this

setting to dedicate an Actor solely to repack requests by disabling the other options (except **Delete**) and disabling repack on the other Actors.

Delete
This parameter defines whether this Actor can be used for requests that involve deleting DIVArchive objects from a disk. You can use this option to isolate an Actor from mass deletion requests (for example, requests issued from the SPM option).

Direct Archive
This parameter defines whether this Actor can be used for direct Archive requests.

Cache Archive
This parameter defines whether this Actor can be used for cache Archive requests. You must disable this option if this Actor has no local cache storage for the temporary storage of the DIVArchive object during a transfer.

First Utilization Date
This is the date the Actor was first put into use.

Advanced Actor Settings

Advanced Actor parameters are displayed on the Actor Advanced Setting Tab in the Actors Panel of the Configuration Utility. Entries are configured and edited on the Actor Settings Entry screen's Actor Advanced Settings Tab. To configure or edit advanced actor parameters, double-click the actor you want to edit to access the settings screen.

The following list describes the parameters on the Actor Advanced Settings Entry screen:

Name
This is the name of the Actor associated with the *Partial File Restore* options. This value is automatically filled in from the Actor settings. If you modify the name here, or in the Actor Settings Screen, it will be modified in both places.

Tape Test Unit Ready Timeout (s)
The time in seconds to wait for a drive to become ready after a tape is mounted. If the drive is not ready within this period, the drive is considered to be not responding.

Profile Read Block Size (B)
The FTP block size used for transfers on profile video servers when reading. The default value (1500) is the best block size to use with GVG profile servers. This value may be different when using other servers. Possible values are between 1500 and 262,144 bytes.

Profile Write Block Size (B)
The FTP block size used for transfers on profile video servers when writing. The default value (32,768) is the best block size to use with GVG profile servers. This value may be different when using other servers. Possible values are between 1500 and 262,144 bytes.

Quantel Rename Clips
Automatically rename clips when restoring them to Quantel.

- Setting this to **N** disables this feature. This is the default setting.

- Setting this to **Y** renames files using the first part of the object name (before the comma) truncated. This is Omnibus renaming.

QT Self-contained Threshold (MB)

When preforming a QuickTime Partial File Restore, the Actor must determine if a clip is self-contained, or not based on the size of the input file. This parameter is a limit in MB. When this limit is exceeded, the Actor considers the clip to be self-contained. The unique objective of this parameter is to prevent the Actor from loading a large self-contained clip into memory. Values range from 10 MB through 100 MB.

Disk FTP Passive Mode

FTP data connections are, by default, created in *Active* mode. The DivaFTP client connects from a random unprivileged port (greater than port 1023). Then it immediately starts listening to the port and sends a PORT command to the FTP server.

When you set this parameter to **Y**, data connections are created in *Passive* mode rather than *Active* mode. In *Passive* mode the DivaFTP client sends a PASV command to the FTP server and the server creates socket, not the client.

Disk FTP Block Size (KB)

This parameter defines how much data the Actor attempts to send and receive using a single system call during FTP transfers.

For example, if the Actor internal buffer size is set to 2 MB, and this parameter is set to 32768 bytes, 64 system calls are required to write a single buffer to a data socket.

Disk FTP Socket Window Size (B)

This parameter adjusts the normal buffer size allocated for output and input buffers. This parameter is internally used to set the send and receive buffers for FTP-managed disk types.

Defining Actor to Disk Connections

After you have configured the Actor definitions, you must define the logical connections (mount points) of the physical disks previously identified during the initial DIVArchive configuration.

If the same resource on a physical disk is to be shared between multiple Actors, and file sharing software has been installed, Oracle recommends that the drive letter or volume of the disk connection configured in each Actor host is identical (for simplicity). Actors retrieve these mount point definitions when the DIVArchive Manager first connects to each of them. Any modifications performed here require the relevant Actor to be restarted.

To edit the parameters, double-click the Actor Name in the *Actor-Disk Connections* frame to open the Add new row in Actor-Disk Connections dialog box. Click the **+** button on the top of the frame to add an Actor-Disk connection.

> **Note:** Multiple selections are available in *Add* mode, but not in *Edit* mode. Nearline storage is used for disk instances created during a Restore or N-Restore request with a **Nearline** QOS when no other disk instances are available.

The following list describes the options on the Add new row in Actor-Disk Connections dialog box:

Disk
Select a physical disk in the drop-down menu for this Actor association. Only entries previously defined in the Disks frame will be displayed. Multiple disks may be selected using the check box next to each disk name.

Actor
Select the Actor for this disk association. Only Actors declared in the Actors frame of the System tab will be listed. The selected disk must be directly accessible by this Actor.

Interface
Select the access method the Actor will use to connect to the disk.

Mount Point
The Mount Point is used with the Interface selection.

Max. Throughput, MB/s
This allows bandwidth throttling of the transfers performed by the Actor. Typically used to load balance transfers with other Actors or non-DIVArchive applications.

When DIVArchive has multiple disks to choose from for object storage, this parameter is the first criteria for disk allocation (that is, the disk with the highest throughput will be used first). The second criterion is the percentage of used capacity of each disk considered.

Access
This defines this Actor's read/write access to the associated logical disk. This allows further granularity in load balancing with other Actors.

Used for
This defines how the associated disk is to be used by this Actor as follows:

Cache Only
DIVArchive will only use this disk for caching operations.

Storage Only
DIVArchive will only use this disk for object storage.

Cache and Storage
DIVArchive will use this disk for both cache and object storage.

Storage and Nearline
DIVArchive will use this disk for both object and Nearline storage.

Cache and Storage and Nearline
DIVArchive will use the disk for cache, object, and Nearline storage.

Actor to Disk Interfaces and Mount Points

The disk interface method and the corresponding mount point in an Actor-Disk connection are determined by how the drive is logically connected and presented to the Actor host computer's operating system. The following sections describe different interface methods and mount point configurations.

Local Interface

This option specifies that *unbuffered I/O* will be used with the disk to maximize transfer performance. Disks that use this option can reside within the Actor host itself (for example, disks to be used for cache purposes), disks connected to the host through either SCSI or Fiber Channel HBAs (for example, in a SAN), or those specified with a Universal Naming Convention (UNC) mount point. Some network drives may actually suffer with this type of interface. In these instances use the *Remote* option instead.

> **Note:**   Windows-based Actors do not support network drives mapped to a Windows drive letter (this is a Microsoft security restriction). Networked disks in Windows must use the *Remote* option instead.

The *Mount Point* is the drive letter or volume of the drive as it appears to the host operating system, plus any additional directory path.

Remote Interface

This interface specifies that *buffered I/O* will be used with the disk, and allows access to disks hosted by another computer using CIFS protocol. This option must be used for networked disks with the Windows Actor Service.

The mount point for a CIFS connection is a UNC path. For example, `cifs://192.168.56.26\shared` or `cifs://user:password@//192.168.56.26\shared`.

Appropriate permissions for any CIFS-based disk must be enabled for the Actor to access the network drive. Otherwise, the disk will be set **Offline**.

> **Note:**   Linux-based Actors do not support UNC paths for CIFS sources and destinations. However, you can define a local path to a mounted SMB share. UNC paths are supported for SMB Source/Destinations and Managed Disks if the UNC path is directly mounted on the Windows Actor.

Oracle Storage Cloud Interface

This interface is used for Oracle Storage and Archive Cloud accounts. See Configuring Oracle Archive Cloud for DIVArchive for configuration information.

BML Interface

This interface enables the Actor to use a SeaChange BML (non-Infiniband Media Libraries) as disk storage. For regular disks, DIVArchive stores objects under multiple subdirectories. The BML however uses a flat file system (that is, no directory structure). DIVArchive automatically incorporates a directory structure into the file name when it is archived to the BML, and removes this addition from the file name as it is restored.

The mount point for the BML option is `bml://IP_Address`. For example, `bml://10.201.10.124`.

FTP Interface

This interface enables DIVArchive to use FTP servers as disk storage using the FTP protocol. Oracle only supports Linux-based FTP servers when operating in a Linux environment, not Windows-based FileZilla and IIS FTP servers. This is because Windows FTP servers cannot handle the large numbers of files.

The mount point must be in the format `ftp://login:password@host/rootdir`.

MetaSAN Interface

You must select this interface when MetaSAN manages the disk volume. By default, DIVArchive Actors preallocate storage on disks to prevent disk fragmentation. MetaSAN implements its own anti-fragmentation mechanisms. Selecting this option will disable preallocation when dealing with this volume.

Simulation Interface

You use this interface when setting up a DIVArchive Simulator. See the *DIVArchive Simulator Operations Guide* for details. This book is only available to OPN partners.

The mount point must be a real path name to a directory on a local disk. When used to store objects, only the file size is recorded to the disk (that is, no content is actually saved). You cannot use a simulated disk as cache for a repack request.

Configuring Actor to Drive Connections

The Data Transfer component of the drives must be configured for use with the Actors separate from the Tape Drive Control configuration for the Robot Manager. You must logically configure of each drive in the Actor-Drive configuration in the database.

The *Actors-Drives* frame is located on the **Drives** tab. The frame displays the current Actor-Drive associations including the Actor Name, Drive Number, and Library location. If a drive is connected to multiple Actors through a SAN, the Actor-Drive mapping must be repeated for each Actor accessing this drive.

You can combine the *Drive Operations* settings and the *Actor Capability* settings to dedicate a drive to a particular set of Actors for specific operations. For example, tape repacking.

To edit the parameters, double-click the Actor Name in the *Actors-Drives* frame to open the Add new row in Actors-Drives Connections dialog box. Click the **+** button on the top of the frame to add an Actors-Drives connection.

Two options are available on the Add new row in Actors-Drives Connections dialog box as follows:

Actor
You select the Actor the drive is connected to from the list. Only Actors already defined in the *Actors* frame of the **System** tab are listed.

Drives
You select the logical drive in the relevant library for this mapping. Only drives defined in the *Drives* frame of the **Drives** tab are listed. You select one or more drives using the check boxes. *Multiple selections are only available when adding an association, not while editing an existing one*.

When you select a different Actor, the drives available for configuration are displayed. If all drives have already been configured for the selected Actor, the *Drives* list is not available and indicates there are no drives available for the selected Actor.

Logging Actor Activity

DIVArchive Actors log all activities during normal operations. The log files are named `actor.log`, or `actor_SERVICE_NAME.log`. The files are stored in the `%DIVA_HOME%\Program\log\actor` folder.

Each DIVArchive Actor also provides additional logging functions for some specific server protocols (for example, the Quantel QCP interface, FTP servers, and Partial File Restore). DIVArchive enables logs by default, and they are unique for each server type. They provide detailed logging information from that protocol to the standard Actor log file.

These files are useful in diagnosing transfer errors with either drives or servers, and particularly for debugging the configuration when a Source or Destination has been added. Oracle Support may request these logs when providing assistance.

Installing and Uninstalling Actor Services in Windows

You use the `actorservice.exe` executable in the Actor `bin` directory to install (or uninstall) the DIVArchive Actor as a service from a Windows command-line prompt.

By default, the Actor Service uses the `actor.conf` file located in `%DIVA_HOME%\Program\conf\actor` folder to define the *Service Name*. If you are installing multiple Actors on a single host, you must create additional Actor configuration files and specify them to the service to create unique instances for each Actor (see Actor Service Management Functions for more information).

See Appendix A for Oracle DIVArchive options and licensing information.

Use the following commands to install or uninstall the Actor Service from the Windows command line:

actorservice -i
Installs the Actor Service using the *SERVICE_NAME* parameter defined in `actor.conf`. If this parameter is undefined, then the service is installed as `DIVArchive Actor - Host_Name`.

actorservice-u
Removes the Actor Service using the *SERVICE_NAME* parameter defined in `actor.conf`. If this parameter is undefined, then the service to be removed is `DIVArchive Actor - Host_Name`.

Installing and Uninstalling Actor Services in Linux

The `divaservice` executable in the Actor bin directory installs (or uninstalls) the DIVArchive Actor as a service from a Linux terminal.

Use the following command sequence to install Actor services:

```
cd/home/diva/DIVA/Program
```

```
./divaservice install actor /home/diva/DIVA/Program/conf/actor/actor.conf
```

Use the following command sequence to uninstall Actor services:

```
cd/home/diva/DIVA/Program

./divaservice uninstall actor /home/diva/DIVA/Program/conf/actor/actor.conf
```

See Installing the DIVArchive Services for more information on using the `divaservice` *command.*

Actor Service Management Functions

When installing or uninstalling additional Actor Services on the same host, you must specify the path to each Actor's configuration file for each instance. You add the `-conf` (or `-f`) command switches when installing the service as follows:

```
actorservice {-i|-u} {-conf|-f} {Path and file name}
```

The command syntax is the same for Windows and Linux. However the path and file name will be different. The following examples install the Actor services for two different Actors on the same host computer. You use the `-u` command switch (instead of `-i` to install) to uninstall these same Actor services.

Check the services applet after installation to verify that each Actor Service was installed correctly.

For example, use the following command in Windows to install the Actor defined by the *SERVICE_NAME* in the `actor1.conf` configuration file:

```
actorservice -i -conf C:\DIVA\Program\conf\actor\actor1.conf
```

Use the following command in Windows to install the Actor defined by the *SERVICE_ NAME* in the `actor2.conf` configuration file:

```
actorservice -i -conf C:\DIVA\Program\conf\actor\actor2.conf
```

Use the following command in Linux to install the Actor defined by the *SERVICE_ NAME* in the `actor1.conf` configuration file:

```
actorservice -i -conf ../../conf/actor/actor1.conf
```

Use the following command in Linux to install the Actor defined by the *SERVICE_ NAME* in the `actor2.conf` configuration file:

```
actorservice -i -conf ../../conf/actor/actor2.conf
```

The following additional command options are also available for the Actor Service:

actorservice debug
Starts the Actor Service in console mode. This is used for troubleshooting.

actorservice version
Displays the DIVArchive Actor Service software release information. You can also use the `-v` switch instead of `version`.

actorservice help
Displays all command line options.

Launching the Actors

Windows DIVArchive Actors no longer start automatically with Windows. You can manage the Actor Services through the Windows Services applet, from a Windows command line, or from Linux terminal.

In Windows, you can locate the Actor Service in the Windows Services applet, right-click the name, and then select the desired management function (**Start**, **Stop**, **Restart**, and so on) from the context menu.

> **Note:** The quotation marks in the following commands must be used when specifying a Windows service with spaces in the name.

You can restart an Actor from a Windows command line or Linux terminal using the following command sequence:

```
net stop "DIVArchive Actor"
net start "DIVArchive Actor"
```

If a *SERVICE_NAME* is specified in the `actor.conf` file (for multiple Actors on a single computer), then you can restart an Actor from a Windows command line or Linux terminal using the following command sequence:

```
net stop "DIVArchive Actor -SERVICE_NAME"
net start "DIVArchive Actor -SERVICE_NAME"
```

> **Tip:** Create a Windows batch file containing these commands and place it on the desktop for easy access.

7

Configuring DIVArchive Managers

This chapter describes DIVArchive Manager configuration and includes the following information:

- Configuration Overview
- Configuring the Local Manager
  - Basic Settings
  - Database Settings
  - Advanced Settings
  - Logging Settings
- Configuring Request Priorities
- Rerouting Destinations (`restore_translations.conf`)
- Controlling the Manager
- Logging Manager Activity
- Confirming System Connectivity
  - Confirming Remote Client to Manager Connectivity
  - Confirming Manager to Actors Connectivity
  - Confirming Manager to Robot Manager Connectivity
- Manager Failover Procedures

Configuration Overview

The Manager module is located in `%DIVA_HOME%\Programs\Manager\bin` in Windows, and in `/home/diva/DIVA/Program/Manager/bin` in Linux, and runs as a service. The static configuration file for the Manager is `manager.conf`. You can typically leave most settings in this file left at the default values. The settings that would normally require updating are highlighted in bold type.

See Appendix A for Oracle DIVArchive options and licensing information.

The following figure is the workflow for installing a DIVArchive Manager:

DIVA_024
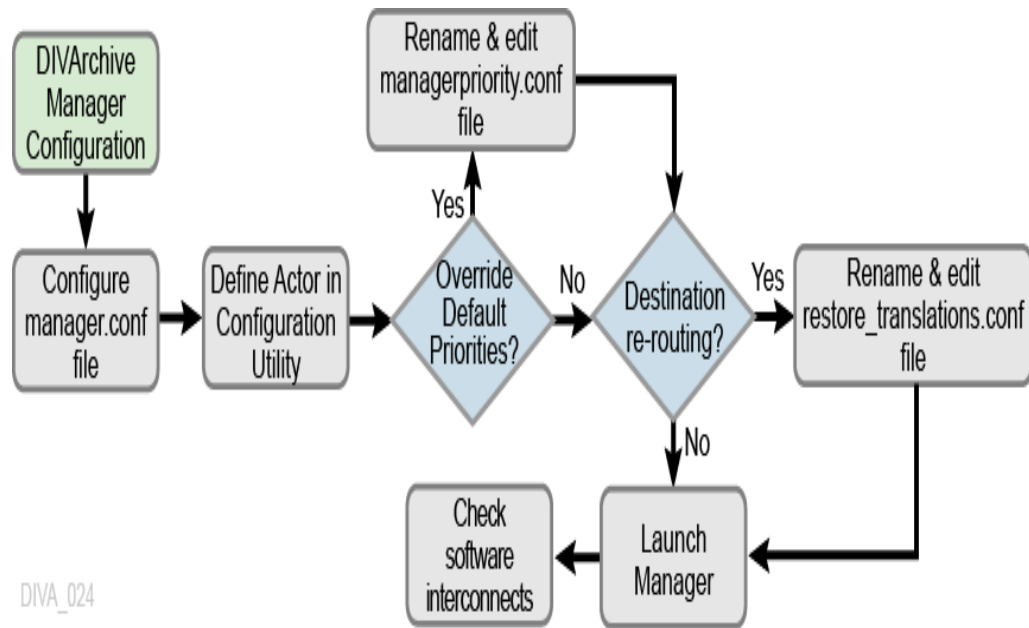
Configuring the Local Manager

The static configuration file in new installations is initially named `manager.conf.ini`. You must remove the `.ini` extension for it to be recognized by the DIVArchive Manager.

The configuration file is divided into five distinct groups; *Basic*, *Database*, *Advanced*, *Logging*, and *Service* settings. You must not modify the *Service* settings section, and therefore, not covered in this manual. *Values defined in this section must only be altered with instruction from Oracle Support.*

Each parameter section in the configuration file contains information on defining that parameter. The information lines are commented out (start with #) and ignored by the Manager. Any parameter definition that is missing the equal sign is also ignored.

Spaces in the parameter settings are significant. Do not put extra spaces before or after the parameter names or their values. If you have trouble running the Manager after configuring the `manager.conf` file, confirm that spaces are not present in any of the parameter values you have defined.

Restarting the Manager can disrupt a live production system. You can make most of the customizations in the configuration file effective immediately using the `restart` command line switch.

If you intend to update your existing DIVArchive system with a newer software release, you must use the `manager.conf.ini` from the new release. You must update the *Basic* and *Database* settings with the values from the old configuration file. The new release configuration file may have additional settings or updates included; this applies to all DIVArchive software modules when installing a release updated.

Basic Settings

Except for the *SERVICE_NAME*, these parameters are always required and must be defined for the Manager to start successfully. These settings define how other DIVArchive software components and DIVArchive API clients connect to the Manager.

> **Note:** These settings are not reloadable while the Manager is running. You must restart the Manager for them to take effect.

The following table describes the *Basic* settings in the `manager.conf` file:

Table 7–1   Basic Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
SERVICE_NAME	Name	You can use this parameter to specify the name of the service. If not defined, the Service Name defaults to DIVArchive Manager.	
DIVAMANAGER_NAME	Name	This is the name this Manager instance uses to identify itself to other DIVArchive Managers sharing its resources. Otherwise, this is arbitrary. It must be unique in a system running multiple Managers except for Main and Backup Managers (configured as a cold standby). In this instance, the names should be identical.	DIVA
DIVAMANAGER_PORT	TCP Port Number	The TCP port other DIVArchive applications, or third party applications using the DIVArchive Client API use to connect to this Manager. If using a Sony library and running the Manager on the same computer as the PetaSite Controller (PSC) software, the PSC Server also uses TCP port 9000, which cannot be modified. In this situation, you must specify another port for the Manager.	9000

Database Settings

These parameters define the location and instance of the DIVArchive Database. Except for the DIVAMANAGER_TNSNAME parameter, you must define all settings in this section for the DIVArchive Manager to launch successfully.

The following table describes the *Database* settings in the `manager.conf` file:

Table 7–2   Database Settings in `manager.conf`

Parameter	Parameter Type	Description	Default

DIVAMANAGER_TNSNAME	Name	The TNS Name of the DIVArchive Schema within the Oracle database. DIVArchive ignores this setting if the `DIVAMANAGER_DBHOST` and `DIVAMANAGER_DBPORT` settings are defined.
		
		This feature requires Oracle 11*g* or higher installed on the host running the Manager. If this setting is defined, the location of the Oracle OCI driver (for example, `ocijdbc11.dll`) must be added to the `wrapper.java.library.path` setting (located in *Service* settings section of the file); otherwise, the Manager will not start as a service.
		
		Example:
		`wrapper.java.library.path=`
		`.;C:\app\oracle\product\11`
		`.1.0\BIN`	

DIVAMANAGER_DBHOST	IP Address or Host Name	This specifies the Host Name or IP Address of the computer containing the DIVArchive Database. If using a host name, this must be present in the `hosts` file on the computer where the DIVArchive Manager is installed.	
DIVAMANAGER_DBPORT	TCP Port Number	The Oracle Listener port configured during the DIVArchive Database installation.	1521
DIVAMANAGER_DBSID	Name	The DIVArchive Database Instance System Identifier (SID) in Oracle where DIVArchive Manager connects. Consult your Site Configuration if you do not know this value.	
DIVAMANAGER_DBUSER	Name	The user name the DIVArchive Manager uses to connect to the DIVArchive Database. This is typically `diva` (case sensitive).	diva
DIVAMANAGER_ DBSERVICENAME	Name	Either this value or `DIVAMANAGER_DBSID` must be set. If both are set, this takes precedence over the SID.	No default value, but `lib5.world` is recommended.

Advanced Settings

You typically leave the parameters in this section are typically left at their defaults. They customize DIVArchive's default behavior for task execution, resource allocation, and the number of connections it will accept from DIVArchive Applications and DIVArchive API Clients. These parameters are normally adjusted or fine-tuned after completing the initial installation of DIVArchive.

Most (but not all) of these settings can be altered while the Manager is running by using the reload option.

The following table describes the *Advanced* settings in the `manager.conf` file:

Table 7–3   Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
DIVAMANAGER_TO_LOWER	**true** or **false**	Sets case sensitivity for DIVArchive. If set to **true**, then all object names, categories and groups will be set to lowercase.	**false**
DIVAMANAGER_MAX_ CONNECTIONS	Number of Connections	Specifies the maximum number of simultaneous client connections the Manager will accept. This includes DIVArchive Actors, Control GUIs, API connections, and support tools.	200
DIVAMANAGER_MAX_ SIMULTANEOUS_REQUESTS	Number of Requests	The maximum number of requests processed by the DIVArchive Manager. When this limit is reached, any further requests will be rejected. *The maximum tested value for this setting is 2000.*	500
DIVAMANAGER_MAX_ INACTIVE_REQUESTS	Number of Requests	Maximum number of inactive requests that cannot find resources examined by the Request Scheduler each time it is activated.	0

Table 7–3   (Cont.) Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
DIVAMANAGER_TYPICAL_ OBJECT_SIZE	Percentage	During operation a DIVArchive Actor retrieves the file size of an object before an archive transfer. This value determines the best location on the tape for the file. Some servers do not indicate the file size of an object before a Direct Archive. Therefore, DIVArchive will use this value as an estimate for tape selection. You must define this setting so that most objects to be archived in the DIVArchive system are below this size.	10 (percent)
DIVAMANAGER_MAX_SPAN_ SEGMENTS	Number	DIVArchive will attempt to span the file across two or more tapes if no more writable tapes with enough free space are available to archive a file. This setting defines the maximum number of tapes across which the object will be spanned.	2 (segments)
DIVAMANAGER_CAPACITY_ LOW_WATER_MARK	Percentage	When the percentage of the total used capacity reaches this amount, periodic warning messages are issued in the Control GUI.	90 (percent)
DIVAMANAGER_ENABLE_ SPANNING_LARGE_OBJECTS	**true** or **false**	Enables spanning of large objects. This parameter overrides SPAN_SEGMENTS if any object in the system is known to be too large.	**true**
DIVAMANAGER_MAX_OBJECTS_ FOR_REPACK	Number	Repacking a tape with many objects can consume resources for a lengthy period without reclaiming a great deal of unused space in the process. This setting prevents this by limiting the selection of tapes in manual and automatic repacks based on the number of objects.	500

Table 7–3   (Cont.) Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
`DIVAMANAGER_STOP_` `IMMEDIATELY_FOR_REPACK`	**true** or **false**	This setting specifies whether to complete any repack requests still running or to terminate them after the Automatic Tape Repack period. If this is set to **true** then repack requests still in progress after the Automatic Repack period will be stopped.	**true**
`DIVAMANAGER_DISMOUNT_` `AFTER`	Time in Milliseconds	This specifies the time in milliseconds to automatically dismount a mounted tape no longer needed by any other request.	`120000` (two minutes)
`DIVAMANAGER_UPDATE_` `PRIORITIES_PERIOD`	Time in Milliseconds	DIVArchive periodically examines all requests in its request queue and increments the request priority. This prevents a condition where low priority requests might be continually superseded by higher priority requests. This setting specifies the period between updates of the queue by the Manager. You set this value to `0` to disable priority updates.	`60000` (one minute)
`DIVAMANAGER_MAX_DELAY_` `BETWEEN_SCHEDULER`	Time in Milliseconds	The maximum number of milliseconds between two Request Scheduler activations when the Manager is constantly busy.	`5000` (five seconds)
`DIVAMANAGER_SCHEDULER_` `AFTER_INACTIVITY`	Time in Milliseconds	The number of milliseconds after which a requested Request Scheduler activation can be launched if the Manager is idle. This duration should be significantly lower than `DIVAMANAGER_MAX_DELAY_` `BETWEEN_SCHEDULER`. *You should not need to modify this value.*	`500`
`DIVAMANAGER_PING_` `INTERVAL`	Time in Milliseconds	This defines the interval in milliseconds between Manager checks to see if the connections to its clients and services are still active (Actors, SPMs, Control GUIs, and so on).	`600000` (ten minutes)

Table 7–3   (Cont.) Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
DIVAMANAGER_EXPORT_ROOT_DIR	Directory Path	The **Export Tapes** command enables the sharing of tapes between two or more separate DIVArchive platforms. This setting defines the root folder for the exported tape's Metadata files. The folder must exist and have write permissions enabled on the host computer where the DIVArchive Manager is running.	Exported
DIVAMANAGER_MAX_RESTORE_SERVERS	Number between 2 and 200	The maximum number of servers allowed in an N-Restore request by a DIVArchive Actor.	5
TAPE_FULL_ON_SPAN_REJECTED	**true** or **false**	If **true**, and spanning is disabled, the Manager marks a tape *full* when spanning occurs.	**false**
DIVAMANAGER_MAX_EXPORT_TAPES	Number between 1 and 25	The maximum number of tapes allowed in an Export Tapes request.	10
DIVAMANAGER_MAX_EXPORT_ELEMENTS	Number between 1 and 100,000	The maximum number of elements that can be exported using the **Export** command.	100000
DIVAMANAGER_MAX_FILES_IN_ARCHIVE	Number between 1 and 1,000,000	The maximum number of files allowed in an Archive request.	1000000
DIVAMANAGER_MAX_FILES_IN_PARTIAL_RESTORE	Number between 1 and 1,000,000	The maximum number of files allowed in a Partial File Restore request.	1000000
USE_IMPROVED_BEST_WORST_FIT_ALGORITHM	**true** or **false**	When a file was archived to tape in earlier DIVArchive releases, the *Best/Worst Fit* algorithm selected the tape with the largest remaining free size. This could result (over time) in a low number of blank tapes for tape repacking, and so on. The current algorithm selects the tape based on the smallest free space and then fills all tapes before using more free tapes.	**true**

Table 7–3   (Cont.) Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
DIVAMANAGER_SITE_ SUPPORT_ENABLED	**true** or **false**	Resources within DIVArchive can be defined by their location. If you set this parameter to **true**, the Manager first tries to perform the request from the sites identified as **MAIN**. If unsuccessful, it retries the request with resources from all other sites. If you set this parameter to **false**, DIVArchive ignores site identification and all site resources are considered equally.	**false**
DIVAMANAGER_CACHE_QOS_ USE_DISK	**true** or **false**	In earlier DIVArchive releases, a Restore request with a *Quality of Service* of **CACHE** or **CACHE and DIRECT** resulted in the tape instance being used as first priority, even if a disk instance existed. This setting instructs DIVArchive to use the disk instance regardless of the QOS method specified.	**true**
DIVAMANAGER_PRIORITY_ TIER	Number between 0 and 100	DIVArchive bases the execution of requests in its request queue by the request priority number. However, there are instances where a request in the queue with lower priority uses a tape that is already mounted. Giving this request priority over others lower in the queue can save a substantial amount of time in tape mount and dismount operations, and help reduce wear and tear on the tape drives. If this setting is enabled, DIVArchive examines the request queue for lower priority requests involving a tape that is already mounted in a drive and adds the number specified here to the request priority. For example, if the request priority is 25, and the *Priority Tier* value is 50, the total request priority is 75.	0 (disabled)

Table 7–3   (Cont.) Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
DIVAMANAGER_ETC_FEATURE	**true** or **false**	This parameter enables the Estimated Time to Complete feature. This function gathers statistics (over time) on the time for completion of all execution states of each DIVArchive request. Setting this value to **true** enables this feature.	**false**
DIVAMANAGER_ETC_ CONFIDENCE_LEVEL	Number	The percentage of Slope Confidence Interval for the simple regression statistical function used in the Estimated Time to Complete feature. DIVArchive ignores this setting if the DIVAMANAGER_ ETC_FEATURE is disabled.	50
DIVAMANAGER_OVERWRITE_ POLICY	Number between 0 and 2	This value determines how DIVArchive handles files that already exist on a Destination server when executing a Restore, Partial File Restore, or N-Restore request as follows: 0 - If the file to be restored to the destination already exists no overwrite will occur. 1 - The Actor does not verify if the files with the same names exist before attempting to overwrite these files. If files with the same names do exist, a backup of the existing files is made before overwriting them. 2 - The Actor attempts to delete and then write to files with the same names.	1
DIVAMANAGER_OVERWRITE_ OVERRIDE	**true** or **false**	Overrides the policy sent by the external application through a request with the policy set in DIVAMANAGER_ OVERWRITE_POLICY.	**false**
ATTEMPT_ACCESS_TO_ OFFLINE_DISK	**true** or **false**	If a disk is offline or not visible to all available Actors, the Manager will automatically terminate a transfer request for objects residing on that disk. If this is set to **true**, the Manager attempts the transfer irrespective of disk status.	**false**

Table 7–3   (Cont.) Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
CHANGE_DISK_STATE_ON_ERROR	**true** or **false**	Defines whether the Manager will automatically vary a disk's status to **Offline** if a transfer error occurs.	**true**
MANAGER_ACTOR_DISK_RETRY_NUMBER	Number	If a disk I/O error occurs during a transfer, this sets the maximum number of transfer retry attempts with alternate Actors that also have access to the disk.	3
DISK_STATUS_POLLING_RATE	Number	This defines the rate in milliseconds in which each disk in the system is polled to obtain its total and remaining free space.	60000 (one minute)
DISK_BUFFER_SPACE	Number	This defines the percentage of the overall space of a disk to keep free.	0.05 (percent)
DISK_CONNECTION_STATE_RESET_DELAY	Time in Minutes	A disk connection will be reset from the **Out of Order** state when a successful access is completed and this amount of time has passed since the connection was set to **Out of Order**.	1.0 (minute)
COMPONENT_SIZE_CONVERSION_TO_KB_RULE	Number	When an element is successfully transferred to tape or disk, the Actor reports the size of the element in bytes. This value is then converted to KB before it is saved to the database. The conversion may be one of three possible values: 1 - KB = (bytes / 1024) + 1 2 - KB = bytes/1024, but if (KB < 1) then KB = 1 3 - KB = Math.ceil(bytes/1024)	3
DIVAMANAGER_MAX_EXCLUDED_INSTANCES	Number	The maximum number of instances excluded from a request that are logged as an event.	3

Table 7–3   (Cont.) Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default

`LOGGING_ROOT_LEVEL`
 	
`LOGGING_TRACE_LEVEL`
 	
`LOGGING_SERVICE_LEVEL`
	DEBUG, **INFO**, **WARN**, **ERROR**, **FATAL**	Defines the level of information written to the respective log files as follows:
		
		■ **DEBUG** - All messages within the Manager are logged. Log files grow rapidly.
		
		■ **INFO** - Information, Warning, Error, and Fatal messages are logged.
		
		■ **WARN** - Warning, Error, and Fatal messages are logged.
		
		■ **ERROR** - Error and Fatal messages are logged.
		
		■ **FATAL** - No messages are logged unless the Manager stops unexpectedly.
			INFO

`DIVAMANAGER_MAX_SPAN_SEGMENTS`	Number	DIVArchive will attempt to span the file across 2 or more tapes if no more writable tapes with enough free space are available to archive a file. This setting defines the maximum number of tapes that the object will span. This setting will completely disable spanning if set to 1 or below. If a span case arises, the Manager retries the request with a new tape using the old Worst Fit algorithm, and the first tape in the attempted span will be marked full. If the second attempt fails, the request will terminate.	`2` (segments)
`DIVAMANAGER_MAX_DB_CONNECTION_ATTEMPTS`	Number	The maximum number of allowable attempts to connect to the database.	`10000`
`DIVAMANAGER_MIN_DB_CONNECTION_PERIOD`	Number	The minimum period (in milliseconds) between connection attempts.	`1000` (milliseconds)
`DIVAMANAGER_MAX_FOLDERS_IN_ARCHIVE`	Number	The maximum number of folders allowed in an Archive request. Performance degradation can occur for values greater than 10000. The maximum value is 10000.	`10000`

Table 7–3   (Cont.) Advanced Settings in `manager.conf`

Parameter	Parameter Type	Description	Default
`DIVAMANAGER_COMPLEX_` `OBJECT_THRESHOLD`	Number	The maximum number of files allowed before an object is classified as a complex object. The maximum value is 10000.	`1000`
`COMPONENT_SIZE_` `CONVERSION_TO_KB_RULE`	Number	This is the *Object Size Conversion Rule*. Use one of the following rules to convert an object component size from Bytes to Kilobytes: `1` - KB = (bytes/1024) + 1 `2` - KB = bytes/1024, but if (KB < 1) then KB = 1 `3` - KB = Math.ceil(bytes/1024)	`3`
`DIVAMANAGER_RESTORE_QOS`	**CACHE_ONLY**, **DIRECT_ ONLY**, **DIRECT_AND_ CACHE**, **CACHE_AND_ DIRECT**, **NEARLINE_ ONLY**, **NEARLINE_ AND_DIRECT**	This identifies the default *Quality of Service* for Restore requests.	**NEARLINE_ AND_DIRECT**

`DIVAMANAGER_TIME_TO_`
`WAIT_FOR_GRACEFUL_`
`SHUTDOWN`
	Minutes	The time to allow for a graceful shutdown to complete.	`1440` (one day)

`ABORT_ARCHIVES_ON_EMPTY_` `FILES`	**true** or **false**	If **true** the Manager terminates an Archive request if it contains an empty file or folder.	**false**

Logging Settings

The following table describes the *Logging* settings in the `manager.conf` file:

Table 7–4   Logging Settings in manager.conf

Parameter	Parameter Type	Description	Default
LOGGING_ROOT_LEVEL LOGGING_TRACE_LEVEL LOGGING_SERVICE_LEVEL	**DEBUG**, **INFO**, **WARN**, **ERROR**, **FATAL**	Defines the level of information written to the respective log files as follows: ■ **DEBUG** - All messages within the Manager are logged. Log files grow rapidly. ■ **INFO** - Information, Warning, Error, and Fatal messages are logged. ■ **WARN** - Warning, Error, and Fatal messages are logged. ■ **ERROR** - Error and Fatal messages are logged. ■ **FATAL** - No messages are logged unless the Manager stops unexpectedly.	**INFO**
LOGGING_MAXFILESIZE	Kilobytes or Megabytes	When the log file reaches this size, a new file is generated and the old one renamed with appropriate time and date stamps. Older log files are subsequently compressed automatically into zip files at one hour intervals.	10 MB
LOGGING_LIFETIME	Hours	This setting defines how long to maintain trace service and zipped log files before deleting them.	50

Configuring Request Priorities

Each request submitted to the DIVArchive Manager is placed in the Manager transfer queue. Request priorities enable DIVArchive to differentiate between important requests, such as Restore requests, over less important events. For example, tape repacks, and so on.

The request priority is a number from zero to one hundred with zero being the lowest priority and one hundred being the highest. The request priority is typically specified when you submit the request (either from the Control GUI or the DIVArchive Client API). You can also alter the priority after you submit the request using the **Change Priority** command.

The default request priority for each request type is preset within DIVArchive. You can override the default priorities (at your discretion) using the following procedure:

1. Navigate to the `%DIVA_HOME%\Program\conf\manager` folder.

2. Rename the `managerpriority.conf.ini` file to `managerpriority.conf`.

3. Edit the `managerpriority.conf` file using a plain text editor (for example, Notepad or Notepad++) to set the desired values for each request type.

4. You must reload the Manager configuration using the **reload** option or restart the Manager for the new settings to take effect.

Regardless of the configured request priority, the Manager will (by default) periodically increment the priority of every request already the request queue. This prevents a condition where a low request priority can be continually overridden by higher priority requests and never executed.

You can disable this feature by setting the `DIVAMANAGER_UPDATE_PRIORITIES_PERIOD` parameter in the Manager configuration file to `0`. You must then reload the Manager configuration or restart the Manager.

Rerouting Destinations (`restore_translations.conf`)

To simplify production workflows, you can configure DIVArchive to automatically override the original destination specified in a Restore, Partial File Restore, or N-restore request based on the object category and original destination. This is called *Destination Rerouting*. Typically, you use this function to enable selective transcoding based on an object category.

You configure *Destination Rerouting* by editing the `restore_translations.conf` file. The file is located in the `%DIVA_HOME%\Program\conf\manager` folder with the Manager configuration file.

The `restore_translations.conf` file is delivered with a `.ini` extension. You must remove the `.ini` extension for this file to be considered by the Manager.

All re-routing entries must be in the following format:

```
DT_Number=Destination_1;Category_1;TranslatedDestination_1
```

The following list describes these parameters:

DT_Number
This must be the first string in the line and start with `DT_Number`. The `Number` can be any value unique among all entries. For example, `DT_0`, `DT_1`, `DT_2`, and so on. Up to three hundred entries are supported.

Destination_1
The destination in a Restore request for this rule to apply.

Category_1
If the Object Category of the request also matches the destination will be re-routed.

TranslatedDestination_1
This is the new destination for the Restore request.

The following example describes how to configure rerouting a destination:

- A video server accepts clips with `Format1`

- The archive contains clips with both `Format1` and `Format2`

- Format 1 objects are in Category 1 (`Cat1`)

- Format 2 objects are in Category 2 (`Cat2`)

You configure this example as follows:

1. Define a Source (`Source1`) that points to the video server with no restore transcode options.

2. Define another Source (`Source2`) that points to the video server with options to transcode to `Format1`.

3. Create a `restore_translations.conf` file containing the following line:

   ```
   DT_0=Source1;Cat2;Source2
   ```

When an object with the category `Cat2` is restored to destination `Source1`, re-route it to destination `Source2` instead. In this manner, the automation can always use `Source1` as the destination in the request.

Objects having a format of `Format1`, which are directly compatible with the video server, will be restored to `Source1` without transcoding.

Objects having a format of `Format2` and a category of `Cat2` match the configuration line and are rerouted to `Source2`. `Source2` has options to transcode them to `Format1` when restoring.

Controlling the Manager

You perform DIVArchive Manager control and management functions from a command prompt on Windows platforms using the `manager.bat` batch file, and from a terminal window using the `manager.sh` script file on Linux platforms. The executable is located in the `%DIVA_HOME%\Program\Manager\bin` folder in Windows, and in the `/home/diva/DIVA/Program/Manager/bin` directory in Linux.

Installing and Removing the Manager Service in Windows

You must first install the DIVArchive Manager as a system service on new systems. You can accomplished this using the `install` (or `-i`) and `uninstall` (or `-u`) command line switches as follows:

manager install
This (or `manager -i`) installs the DIVArchive Manager service set by the *SERVICE_NAME* parameter defined in `manager.conf`. If this parameter is undefined, the service is installed as `DIVArchive Manager`.

manager uninstall
This (or `manager -u`) removes the DIVArchive Manager service set by the *SERVICE_NAME* parameter defined in `manager.conf`.

In the Windows Services applet, confirm that the DIVArchive Manager service is installed correctly. If you must change the service name, uninstall the existing service *before* editing the `manager.conf` file. Then reinstall the service after changing the service name.

The default path to the m,anager.conf file is `%DIVA_HOME%\Program\conf\manager`.

You can identify a specific configuration in the command line if you require using an alternate file using the `-conf` or `-f` switch as follows:

```
manager install -conf [configuration file]
manager uninstall -conf [configuration file]
```

Installing and Removing the Manager Service in Linux

The `divaservice` executable in the Manager `/home/diva/DIVA/Program` directory installs (or uninstalls) the DIVArchive Manager as a service from a Linux terminal. *See Installing the DIVArchive Services for more information about using the `divaservice` command.*

Use the following command sequence to install the Manager service:

```
cd/home/diva/DIVA/Program

./divaservice install manager /home/diva/DIVA/Program/conf/Manager/manager.conf
```

Use the following command sequence to uninstall the Manager service:

```
cd/home/diva/DIVA/Program
```

```
./divaservice uninstall manager /home/diva/DIVA/Program/conf/Manager/manager.conf
```

Managing the Manager Service

You can manage the Manager Service using the following command line switches after the service is installed:

manager start
This switch starts the Manager service (if stopped).

manager stop
This switch stops the Manager service (if running).

manager shutdown
This switch finishes currently running requests and stops accepting new requests, then it stops the DIVArchive Manager service (if running).

manager restart
This switch stops and subsequently starts the Manager service.

manager reload
Some changes in the Manager configuration files take effect after reloading the Manager. This switch reloads the `manager.conf`, `managerpriority.conf`, and `restore_translations.conf` files from the default path (`%DIVA_HOME%\Program\conf\manager`).

Use the following command to reload the Manager using a different configuration file:

```
manager reload -conf [configuration file]
```

manager status
This switch displays the current status of the Manager service (running or not running).

manager dump
This switch requests a system dump from the Manager service.

manager version
This switch (or `manager -v`) displays the Manager service release information and then exits.

manager help
This switch (or `manager -h`) display all command line options and then exits.

Logging Manager Activity

The DIVArchive Manager keeps detailed logs of its operations and stores them in the `%DIVA_HOME%\Program\log\manager` folder. The logs are used for troubleshooting and diagnostics purposes, and may be requested by Oracle Support.

The logging settings in `manager.conf` determine the level and quantity of information captured in each log file. If you must alter the settings, you can make the changes effective immediately using the `manager reload` command, or (in DIVArchive 7.5) change them dynamically from the Control GUI. See the *Oracle DIVArchive Operations Guide* in the *Oracle DIVArchive 7.5 Core documentation* library for more details.

Class-level logging is supported through the `manager.classLog.properties` file. Any class set to one of the following values will log at the specified logging level:

- **TRACE**
- **DEBUG**
- **INFO**
- **WARN**
- **ERROR**
- **FATAL**

New statical data is generated every five minutes that lists various Manager performance related metrics, and collected in a statistics folder.

Class Level Logging overrides the settings in the `manager.conf` file. The purpose of Class Level Logging is to provide the capability for the Oracle Development Team to gather additional logging information when debugging issues. *The defaults should never be changed without consulting the Oracle Development Team*.

The following is a partial sample from a `manager.classLog.properties` file, and not the complete file:

```
# -------------------------------------
# DIVA Manager Class Level Log Properties
# -------------------------------------

com.storagetek.ComponentNames=WARN
com.storagetek.compression.CompressionUtilities=WARN
com.storagetek.diagnostics.Assert=WARN
com.storagetek.diagnostics.AssertFailure=WARN
com.storagetek.diagnostics.EventsQueue=WARN
com.storagetek.diagnostics.InternalFailureException=WARN
com.storagetek.diagnostics.QueueingAppender=WARN
com.storagetek.diagnostics.Trace=WARN
```

After logs have reached the size defined by *LOGGING_MAXFILESIZE* in `manager.conf` they are renamed with date and timestamps, compressed (zipped), and a new file is started (named `manager.trace`). The `manager.trace` file is the log file currently being written to by the Manager.

Confirming System Connectivity

After the DIVArchive Manager has been successfully configured and launched you must check that the Manager can successfully be connected to by other DIVArchive clients (for example, the Control GUI). Also, the Manager itself must be able to connect to the configured Actors and, if installed, Robot Managers.

Confirming Remote Client to Manager Connectivity

This short test establishes whether the Manager is configured correctly and accepting remote connections from clients:

1. Launch the DIVArchive Control GUI from a remote client (that is, not on the same host computer as the DIVArchive Manager).

2. Click the **Menu Orb** on the top left of the Control GUI.

3. Click **Connect**.

4. Enter the *IP Address* and *TCP Port* of the Manager in the Connect to the Manager dialog box.

5. Click **Connect**.

6. A successful connection will be indicated by a *Connected* status in the Control GUI notification area (at the bottom of the screen).

Confirming Manager to Actors Connectivity

This short test establishes whether the Manager can connect to all Actors in the system. This test assumes all Actors have been configured correctly and are online.

With the Control GUI still open, click the **Actors** icon in the **Home** tab on the icon bar to display the *Actors* view.

Confirm that the Manager has established a connection to all configured Actors, and troubleshoot if necessary.

Confirming Manager to Robot Manager Connectivity

This short test establishes whether the DIVArchive Manager can connected to each configured DIVArchive Robot Manager. This test assumes the following:

■ All DIVArchive Robot Manager are configured correctly.

■ Each DIVArchive Robot Manager is running.

■ All libraries are loaded with tapes.

■ Any library management software (for example, ACSLS) is running, and the library is set to **Online**.

■ Manual operation has been confirmed successfully with the DIVArchive Robot Manager Client Tools.

Use the following procedure to confirm connectivity:

1. Click the **Tapes** icon on the **Home** tab to display the *Tapes* view.

2. Take note of the *ACS* and *LSM* number for each tape to test each particular library.

3. Right-click a tape for each *ACS* and *LSM* to test and click **Eject Tape** from the resulting menu.

4. Click the **Manager** icon on the **Home** tab to display the *Manager Current Requests* view.

5. Double-click the Eject Tape request entry to check if an error was encountered during request execution.

Manager Failover Procedures

> **Caution:** The procedures in this section are critical and sensitive. They should only be performed under the control of Oracle Support.

You use the following procedures to switch to a Backup Manager (if possible) if a Manager failure occurs.

You perform the following sequence of steps on the *Main Manager*:

1. Attempt to stop the DIVArchive Manager service if it is still running.

2. Execute the **DIVA DB Full backup** scheduled task.

3. Execute the **DIVA DB Backup sync** scheduled task.

4. Shutdown the DIVA Manager server as cleanly as possible.

You perform the following sequence of steps on the *Backup Manager*:

1. Change the IP address to the *Main Manager* computer's address.

2. Restart the *Backup Manager* computer.

3. Use the following command sequence to recover the database:

    1. Execute `C:\app\oracle\admin\rman\bin\restore_lib5_from_mgr1_to_mgr2.bat`.

    2. Enter `0` for *Automatic Restore* and wait for completion.

    3. Enter 0 for *Full Backup*.

    4. Enter `Q` to quit.

4. Start the DIVArchive services; Manager, RobotManager, Storage Plan Manager, DFM, and so on depending on your system configuration.

If all steps completed successfully, the original *Backup Manager* computer is now the *Main Manager* computer, and running the DIVArchive Manager. You can now repair or replace the original *Main Manager* computer.

8

Configuring DIVArchive Checksum Support

This chapter describes configuring DIVArchive Checksum Support and includes the following information:

■ Configuration Overview

■ Global Checksum Parameters

■ Configuring Checksum Support for Sources and Destinations

■ Configuring Checksum Support for Arrays and Disks

■ Configuring Checksum Support for Groups

■ Configuring Checksum Support for Actors

■ AXF and TEXT Genuine Checksum Modes

   – Configuring AXF Genuine Checksum Mode

   – Configuring TEXT Genuine Checksum Mode

Configuration Overview

You configure the Checksum Support functions through the Configuration Utility using the *Engineer* account. The following sections describe how to adjust the settings for each option.

Global Checksum Parameters

You must use the *Engineer* account in the Configuration Utility to access and adjust the *Global Checksum Parameters* located under the **Manager Setting** tab. Each of the global parameters affects all Checksum Support settings throughout the system. The following options are available:

Manager: Checksum feature is enabled
This setting enables (check box selected) or disables (check box deselected) the Checksum Support features throughout DIVArchive. The default setting is enabled (selected).

Manager: Default Checksum Type
There are several checksum algorithms supported by the system including MD2, MD5, SHA, SHA1, MDC2, and RIPEMD160. DIVArchive uses MD5 as the default checksum.

Each checksum type is associated with an ID Number. you use the menu list to change the default type and select the type of checksum desired.

The ID Number identifies the Checksum Type requested in the configuration as follows:

- MD2 is ID Number 1

- MD5 is ID Number 2

- SHA is ID Number 3

- SHA-1 is ID Number 4

- MDC2 is ID Number 5

- RIPEMD160 is ID Number 6

Manager: Number of retries following failed checksum
This parameter sets the number of times the system will retry the operation after a failed checksum. The default setting is one retry. Enter the number of retries allowable for your data and system in the *Manager: Number of retries following failed checksum* field. *Oracle recommends leaving this setting at the default value.*

Manager: Select different drive per retry on failed checksum
This parameter distinguishes whether the retry (after a failed checksum) is attempted on the same drive (check box deselected), or if the system should attempt the operation using a different drive (check box selected). The default setting for this parameter uses the same drive (check box deselected).

Configuring Checksum Support for Sources and Destinations

You adjust the Checksum Support configuration for sources and destinations through the Configuration Utility **System** tab. In the *Sources and Destinations* frame, double-click the Source or Destination requiring Checksum Support configuration. The Edit Source and Destinations Entry dialog box appears with several Checksum Support configuration options. These options are mainly associated with the Genuine Checksum Type.

The following list describes the options available:

External Checksum Source
You must use the *External Checksum Source* (**Yes** option) for the system to read the Checksum from the external source providing the file. This initiates an immediate checksum calculation to compare the checksums and verify the initial transfer. Selecting the **No** option disables Genuine Checksum support from external sources.

Checksum Type
You use the menu list to select the *Checksum Type*. All supported checksum types are listed. The *Checksum Type* and *GC Mode* (see the following description) must match the settings implemented at the Source.

The Genuine Checksum is only used for the first verification. Therefore, the checksum type selected is only used once and then discarded. Beyond the initial use of the selected checksum type (after this transfer), the default type is used.

GC Mode
You use the menu list to select the *Genuine Checksum Mode*. This notifies the Actor of the format of the files that contain the checksum data.

Verify Following Archive (VFA)
When *Verify Following Archive (VFA)* is turned on (check box selected), performing the initial transfer from the source results in a read-back operation. Therefore, the data

is read twice for verification. After the data is read twice, the two checksums are compared. If they are the same then verification is complete. If they are not identical then verification has failed.

Verify Following Archive is not compatible with Genuine Checksum (GC) or complex objects. There is no need to use VFA when GC is being used because the checksum is already verified. The Genuine Checksum must be turned off to gain access to the VFA check box. If GC is turned on, the check box will be grayed out and not selectable.

Verify Following Restore (VFR)

When *Verify Following Restore (VFR)* is turned on (check box selected), performing the final transfer to the destination results in a read-back operation. Therefore, the data is read twice for verification. After the data is read twice, the two checksums are compared. If they are the same then verification is complete. If they are not identical then verification has failed. The setting of GC has no bearing on the VFR setting.

Verify Following Restore is not compatible with complex objects or the -axf option. Verify Following Restore was designed to read back the restored content from a video server to confirm that it is not corrupt. Using the **-axf** option does not create a checksum verifiable restore. It creates an object export that is encompassed in an AXF wrapper. *The VFR and -axf options are mutually exclusive and should not be part of the same workflow.*

Configuring Checksum Support for Arrays and Disks

You configure Checksum Support for Arrays and Disks through the Configuration Utility **Disks** tab. You can turned on or off Verify Write (VW) functionality either on an array basis or disk by disk.

VW applies when you write to the final storage location in DIVArchive. When turned **ON**, the system will perform a read-back of what was just written and compare the checksums for verification.

The *VW* column in both the *Arrays* frame and *Disks* frame indicates whether the Verify Write function is on or off for the particular array and disk. The default setting is **OFF**.

If there is nothing defined in the *VW* column on the *Disk* frame the system will use the setting defined in the *Array VW* column.

To override the setting defined in the *Array VW* column for a specific disk, you select the disk requiring configuration in the *Disks* frame and click **Edit** located at the top of the frame.

When the Edit Disks Entry dialog box appears, use the *Verify Write* menu list to select **ON**, **OFF**, or **NONE** (blank selection). If **NONE** is selected, Verify Write uses the setting identified in the array for this particular disk.

The selection made in the Edit Disk Entry dialog box is reflected in the *Disks VW* column.

Configuring Checksum Support for Groups

You can also configure Verify Write for Groups. The *VW* column displays in the *Groups* frame of the Configuration Utility. *This is the only place where configuration of Verify Write is available for the Groups.*

Similar to the configuration for disks, select the group requiring configuration. Click **Edit** and select **ON** or **OFF** using the *Verify Write* menu list. Your selection is reflected in the *Groups VW* column.

When DIVArchive writes a file to a particular group, the setting for that group is applied to the file. The default setting for groups is **OFF**.

Configuring Checksum Support for Actors

You can configure *Verify Tape* for Actors. Similar to the configuration for disks and groups, you select the Actor requiring configuration, click **Edit**, and then select **Yes** or **No** using the *Verify Tape* menu list.

This setting defines whether the Actor is automatically selected for the Verify Tape workflow. By default, all Actors are included, but you can exclude if necessary.

AXF and TEXT Genuine Checksum Modes

There are two additional Genuine Checksum modes as follows:

AXF Genuine Checksum Mode
This mode enables DIVArchive to archive all files and subfolders in a specified AXF file while comparing their checksum values against known values stored in the AXF file. This workflow is typically combined with a Restore request with **-axf** in the *Request Options*.

TEXT Genuine Checksum Mode
This mode enables DIVArchive to archive all files and subfolders in a specified folder while comparing their checksum values against known values stored in an external checksum file.

Configuring AXF Genuine Checksum Mode

There are specific requirements and limitations when using the AXF Genuine Checksum Mode as follows:

- The AXF file containing the files to be archived must contain checksum information for each file.

- The checksums must be the expected type as specified in the configuration.

- This workflow only works with AXF requests generated by DIVArchive.

- Verify Following Restore (VFR) is not compatible with the -axf option.

  VFR was designed to read back the restored content from a video server to verify it has not been corrupted. Using the **-axf** option does not create a *real* restore, rather an object export in an AXF wrapper. *These options are mutually exclusive and should not be part of the same workflow*.

DIVArchive Configuration Utility Settings
Use the following procedure to configure AXF Genuine Checksum Mode in the DIVArchive Configuration utility:

1. Create a new Source/Destination entry with the *Source Type* set to either **DISK**, **FTP_STANDARD**, or **EXPEDAT** as appropriate.

   If you are required specify an appropriate *Root Path*, this path along with the input files specified during the Archive request, is used in determining the location of the checksum file.

   For example, if the *Source Type* is **DISK**, you can set the *Root Path* to D:\root. If the *Source Type* is **FTP_STANDARD**, you can set the *Root Path* to /root.

2. Set the *External Checksum Source* to **YES**.

3. Set the *Checksum Type* to the expected checksum type (for example, **MD5**).

4. Set the *GC Mode* to **AXF**.

5. Click the **OK** button.

6. Notify the Manager of the configuration by selecting **Tools**, then **Notify Manager** from the menu.

Configuring TEXT Genuine Checksum Mode

There are specific requirements and limitations when using the TEXT Genuine Checksum Mode as follows:

- A checksum file must be present in the folder specified by the *Root File Path*.

- Checksum files must end with a `.md5` file extension.

- The checksum file name (excluding the extension) must be associated with the folder name that contains all files to be archived. This folder must exist.

  For example, if the checksum file is `D:\Data\Video\NewTitle.md5`, then all files located in the `D:\Data\Video\NewTitle` folder will be archived.

- The checksum file must be present in the folder parent to the folder specified by the *Root File Path*.

- For a file to be archived with the Genuine Checksum value, the file must be referenced with a corresponding checksum within the checksum file.

- Absolute path names are supported on both Windows and Linux to a maximum of 4000 characters. Relative path names are limited to 256 characters on Windows systems (only).

- Linux paths, file names, and commands are case-sensitive.

- Only ASCII, non-UTF-8 encoded checksum files are supported.

- The format of the checksum file is that each line begins with an MD5 checksum, followed by 2 spaces, and then the file path to the referenced file.

DIVArchive Configuration Utility Settings

Use the following procedure to configure TEXT Genuine Checksum Mode in the DIVArchive Configuration utility:

1. Create a new Source/Destination entry with *Source Type* set to either **DISK** or **FTP_STANDARD**.

2. Specify an appropriate *Root Path*. This path, along with the input files, specified during the Archive request is used in determining the location of the checksum file (see Selecting the Root File Path for further details).

   For example, if the *Source Type* is **DISK**, you can set the *Root Path* to `D:\Data`. If the *Source Type* is **FTP_STANDARD**, you can set the *Root Path* to `/Data`.

3. Set the *External Checksum Source* to **YES**.

4. Set the *Checksum Type* to **MD5**.

5. Set the *GC Mode* to **TEXT**.

6. Click the **OK** button.

7. Notify the Manager of the configuration by selecting **Tools**, then **Notify Manager** from the menu.

Selecting the Root File Path

The *Root File Path* must point to the folder containing the checksum file. Therefore, you must set the correct file and folder paths in the Source/Destination and Archive request form so the checksum file can be located. For example, if the checksum file is located in `D:\Data\Video\NewTitle.md5` (or `/Data/Video/NewTitle.md5` for FTP type), you set the appropriate file and folder paths as follows:

Table 8–1   Sample Root File Paths for Disks

Source/Destination (Root Path)	Archive Request (File Path Root)	Archive Request (Files)
D:\	Data\Video\NewTitle	*
D:\Data	Video\NewTitle	*
D:\	Data\	Video\NewTitle*
D:\		Data\Video\NewTitle*

Table 8–2   Sample Root File Paths for FTP

Source/Destination (Root Path)	Archive Request (File Path Root)	Archive Request (Files)
/	Data/Video/NewTitle	*
/Data	Video/NewTitle	*
/	Data/	Video/NewTitle/*
/		Data/Video/NewTitle/*

9

Installing and Configuring Transcoders

This chapter describes installing and configuring transcoders for DIVArchive and includes the following information:

- Transcoder Overview
- Upgrading from Telestream Vantage 5.0 and Earlier
- Installing Telestream Vantage
- Configuring DIVArchive and Transcoders
  - Installing the Telestream License
  - Preparing a Mount Point for Linux-based Actors
  - Configuring the Transcoder and Actor on a Single Computer
  - Configuring the Transcoder and Actor on Separate Computers
- Configuring Telestream Vantage
  - Creating the Output Path
  - Creating a Minimum Vantage Workflow
  - Creating a Complex Vantage Workflow
- Configuring Transcoders
- Configuring Sources and Destinations

Transcoder Overview

The following instructions are directed toward servers running the Windows Server 2012 R2 SP1 operating system. Linux-based Actors only support Telestream Vantage for transcoding operations.

Upgrading from Telestream Vantage 5.0 and Earlier

Upgrading from 5.0 or earlier releases of Vantage requires uninstalling and reinstalling the Vantage software. Refer to the *Vantage 6.3 Installation Guide* for details on the uninstall procedure.

Installing Telestream Vantage

Oracle recommends that no anti-virus software is installed on the Vantage servers. Use the following procedure to install Vantage 6.3:

1. Download the Vantage 6.3 release from Telestream.

2. If you are uncertain of how to install the software, refer to the *Quick Start Instructions* in the downloaded file.

3. Install .NET 3.5 SP1, if not already installed, on the host computer that will be running the Vantage Database server.

4. Install QuickTime 7.6.9 if not already installed.

5. Install the Desktop Experience option. This is located in the Server Manager under **Features**.

6. Install the VantageDatabaseSetup_SQL2008_4.2.286.100451.exe, accepting the default settings.

7. Execute the Vantage_6.3_Setup.exe.

8. Select the **Install Product(s)** option.

9. Ensure the following options are selected:

    - *Transcode/Transcode Pro*

    - *Web Applications*

    - *Workflow Portal Application*

    - *Vantage Domain Database*

10. Enable any other options required for your installation.

11. Complete the installation.

Installing the Telestream License

Use the following procedure to install the Telestream license after the software is installed:

1. launch the Vantage Workflow Designer.

2. If you are prompted to select a *Domain*, select the local computer.

3. If you are prompted for a *Category* click **Cancel** (for now).

4. Click **File**, and then **Add/Update License**.

Vantage is now installed and you can continue with configuring it to work with DIVArchive.

Oracle recommends importing sample workflows in the Vantage Workflow Designer. You can view a demonstration at http://www.telestream.net/vantage/demos.htm.

Configuring DIVArchive and Transcoders

The following instructions identify the configuration of DIVArchive and transcoders to enable operation together. Starting with DIVArchive 7.3, it is no longer required to have Actor installed on the same computer as the transcode service.

A transcoder is no longer coupled to a single Actor. You select the transcoder after you select the Actor. Therefore, you no longer define a **LOCAL** transcode Actor as a destination. A **LOCAL** Actor destination is dynamically and temporarily (only in memory, not stored in the database) created for the Actor that you chose as part of resource selection. *The Actor column was removed from the Transcoders area in the Configuration Utility*.

The transcoder server and cache location is now embedded in the *Working Directory* on the Edit Transcoders Entry screen in the following format:

```
[actor:actor_name,actorPath:actor_transcoder_cache_path,transcoder:trancoder_ip_
address],cifs://user_name:password@\\transcoder_cache_ip_address\transcoder_cache
```

For example:

```
[actor:actor_
001,actorPath:/tmp/vantagecache,transcoder:10.145.40.81],cifs://user:password@\\10
.145.40.81\VantageIn
```

The following rules apply:

- The order of the `actor`, `actorPath`, and `transcoder` settings is important. The order of the parameters must be `actor`, followed by `actorPath`, and finally followed by `transcoder`.

- Multiple transcoders are not supported for Flip Factory. They are only supported for Vantage.

- Linux-based Actors only support Telestream Vantage for transcoding operations.

- If the `transcoder` parameter is not specified with the transcoder IP address, a local address of `127.0.0.1` is assumed.

  For example:

  ```
  [actor:actor_001_std,transcoder:127.0.0.1],d:\diva\local
  ```

- If the `actor` parameter is not specified with an Actor name, the transcoder is presumed to not be mapped to a specific Actor.

- The `transcoder_cache` folder is the location where both the Actor and Vantage use to perform the entire transcode operation. Because Vantage runs in the Windows platform, a CIFS formatted UNC path that is Windows compatible represents the `transcoder_cache` share folder. Vantage will use this path for transcoding.

- If the `actorPath` parameter is not specified, Actor will use the same CIFS formatted UNC path.

- An `actorPath` parameter *must be specified* if you use a Linux Actor for transcoding with Vantage Transcoder as follows:

  ```
  [actor:actor_name, actorPath:actor_path,transcoder:trancoder_ip_
  address],cifs://username: password@\\transcoder_cache_ip_address\transcoder_
  cache
  ```

- Because UNC paths are not accessible by Linux, the Actor will be unable to read from or write to the `transcoder_cache` folder. In this scenario, you must provide an alternate path to the Actor using the `actorPath` parameter. The path is a configured mount point enabling the Linux Actor to transfer content to and from an SMB network share (transcoder cache).

- The original method of configuring a transcoder to a local Actor is still supported for legacy purposes

- The original method of configuring *Local Sources/Destinations* tied to Actors is still supported so legacy configurations will continue to function.

Preparing a Mount Point for Linux-based Actors

Linux-based Actors must have access to the transcoder cache folder through a local mount point. Use the following procedure to create a mount point to the remote transcoder cache:

1. Open a terminal window and execute the `id` command to confirm that you are logged in under the same user account that runs the Actor (typically `diva`) as follows:

    ```
    [diva@Linux018 actor]$ id
    ```

    The response will look similar to the following. Confirm the User ID (`uid`) and Group ID (`gid`), and use these values in the mounting operation. In the example they are both `1000`.

    ```
    uid=1000(diva)gid=1000(diva)
    groups=1000(diva),10(wheel),30(tape),54321(oinstall),54322(dba)
    context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
    ```

2. Execute the following command to create your local mount point directory (for example, `/tmp/vantagecache`). You may need to log in as the `root` user depending on where the directory is located.

    ```
    mkdir /tmp/vantagecache
    ```

3. Execute the following command to create a local mount point (for example, `/tmp/vantagecache`) to the network share of the transcoder cache. Enter the appropriate share authentication information including the `user_id#`, `group_id#`, `remote_transcoder_cache_ip_address`, and `remote_transcoder_cache_path`.

    ```
    mount -t cifs -o username={user_name},password={password},uid={user_
    id#},gid={group_id#} //transcoder_cache_ip_address/remote_transcoder_cache_
    path /tmp/vantagecache
    ```

4. Set the `actorPath` parameter to `/tmp/vantagecache` when configuring the transcoder settings.

The following sections describe general transcoder configuration.

Configuring the Transcoder and Actor on a Single Computer

Use the following procedure to configure Vantage transcoders when the Actor is on the same computer as the transcode service:

1. Create a cache folder on the Actor computer. For a Vantage transcoder you could use `M:\VantageCache`.

2. Add the transcoder in the DIVArchive Configuration Utility. with the following settings:

    -   *Transcoder Type*: `vantage`

    -   *Working directory*: `M:\VantageCache`

    -   Leave the remaining options at the default settings.

3. Open the DIVArchive Configuration Utility.

4. Navigate to the *Transcoders* frame on the **System** tab.

5. Ensure that the DIVArchive Transcoder configuration's *Simul Transcodes* value is less than or equal to the corresponding *Vantage Session Limit* value.

6. Open the Vantage Management Console.

7. Click **Services** in the left navigation tree.

8. Locate the transcoder you are configuring in the right frame and then right-click the transcoder name.

9. Select **Enter Maintenance Mode** from the context menu.

10. Click **Service Limits** on the **Setup** tab in the bottom frame.

11. Confirm the *Session Limit* and the *Target Resource Usage* parameters are set correctly for your environment, and adjust as necessary.

Configuring the Transcoder and Actor on Separate Computers

Use the following procedure to configure Vantage when the Actor is on a different computer than the Vantage Transcode service:

> **Caution:** The cache folder must be located on a computer accessible by the Vantage SDK computer through a shared Windows path.

1. Create a cache folder on the remote computer. In the example `M:\VantageCache` is used.

2. In Windows, share this folder on the network and set the required access credentials.

3. Authorize the Vantage transcoder to access the shared Vantage Cache folder.

4. Open the Vantage Management Console on the Vantage SDK computer.

5. Navigate to the **Settings & Options** screen using the left navigation tree.

6. Click the **Authorization** tab.

7. Add a new entry with the *Username*, *Password*, and *Folder*. For example, `\\10.145.50.81\VantageCache` is the Windows UNC path for the shared Vantage Cache folder.

8. Open the DIVArchive Configuration Utility.

9. Navigate to the *Transcoders* frame on the **System** tab.

10. Add the transcoder to the DIVArchive Configuration Utility with the following settings:

    ■ *Transcoder Type*: vantage

    ■ Set the *Working Directory* as follows:

       – Use a CIFS UNC path pointing to the IP address of the Vantage Cache computer. Include the required authentication information for the shared Vantage Cache folder.

       – Include the path to the shared Vantage Cache folder.

       – If the Actor is Linux-based, you must set the mount point to enable the Actor to access the Vantage Cache. Set the `actorPath` parameter to point to the mount point.

       – If the Vantage Cache is located on a different computer than the Vantage SDK service (different IP address), you must tell the Actor the IP address

where the transcoder service is located. Set the transcoder parameter to point to the address of the Vantage SDK service computer.

■ Leave the remaining options at the default settings.

The following is an example Working Directory entry:

```
[actorPath:/tmp/vantagetest,transcoder:10.145.40.81],cifs://user:pass@\\10.145
.40.81\VantageCache
```

Configuring Telestream Vantage

The following sections describe only the configuration for the Vantage transcoder.

Creating the Output Path

Use the following procedure to create the output path in Vantage:

1. Open the Vantage Management Console and connect to the local computer.

2. In the left navigation tree, navigate to **Workflow Design Items**, **Variables**, **Create New Variable**.

3. Use the menu list to set the *Select the variable type* parameter to **Path**.

4. Click **OK**.

5. At the bottom of the screen, update the *Name* field to OutputPath.

6. Click the **Save** icon to save the variable.

Creating a Minimum Vantage Workflow

Use the following procedure to create the minimum Vantage workflow. First, you create the workflow and link the **Receive** and **Flip** together as follows:

1. Open the Vantage Workflow Designer.

2. Create a **New Category**.

    In the example the *Name* is TESTMINWorkflow.

    > **Note:** No spaces or special characters are allowed in the category name.

3. Create a **New Workflow** and enter a name for it in the *Enter a name* field.

4. Select the *Category* for the workflow from the *Select a category for the new workflow* list.

5. Optionally, you can enter a description in the *Enter a description* field if desired.

6. If desired, set the number of hours for the workflow to expire, and select the *Expire after* check box.

7. Click **OK** to save the workflow.

8. Click the **Common** icon, and the click **Receive**.

9. Click the **Transcode** icon, and then click **Flip**.

10. To link the *Receive* and *Flip* together, click the **Receive yellow dot** and drag it to the **Flip yellow dot**.

Next, you configure the **Flip** options as follows:

1. Right-click **Flip** to configure the Flip options. For this example a media file is being configured using the following settings:

   ■ *Encoder*: `Apple 3GP`

   ■ *Input media file nickname*: `Original`

   ■ *Output media file nickname*: `Mobile`

2. Expand the **Output Location** section.

3. Select the *Path* option, and then enter, or browse, to select the output path (for example, `E:\VantageStore`).

4. Use the menu list to select the *Collision Resolution*. This identifies what the software will do if there is an existing file in the output path with the same file name. Initially set the *Collision Resolution* field to **Overwrite**.

5. Click **Save** to save the configuration.

Next, you configure the Receive options as follows:

1. Right-click *Receive* to configure the Receive options.

2. Click the *Media Files* list and choose **Vantage Proxy**.

3. Click **Save** to save the changes.

When you are finished, click **Release** to enable DIVArchive to use the workflow.

Creating a Complex Vantage Workflow

This Vantage complex workflow example was created for documentation purposes; however it has not been tested with actual media files. Use the following procedure to create the complex Vantage workflow:

1. Open the Vantage Workflow Designer.

2. Navigate to **File**, and then create a **New Category**. For this example the *Category* created is named `TESTComplex`.

3. Navigate to **File**, and then **Import Workflow**.

4. Browse and select `C:\Program Files (x86)\Telestream\Vantage\Samples\Analysis\Smart SD and HD Transcoding.xml`.

5. Specify the category created in Step 2.

6. Oracle recommends changing the *Workflow Name* to match the *Category*. *No spaces or special characters are allowed*.

7. Delete the *Watch* and replace it with *Receive*.

8. Configure *Receive* and set *MediaFiles* to **Original**.

9. Link *Receive* with *Identity*.

10. Delete *Deploy*.

11. Configure both Flip Factories.

- Change the *Output Location* to **Path** and then enter, or browse, to select the output path.

- Change the *Collision Resolution* to **Overwrite**.

12. Click Release to enable DIVArchive to use the workflow. In this example, the workflow should look like the following figure.
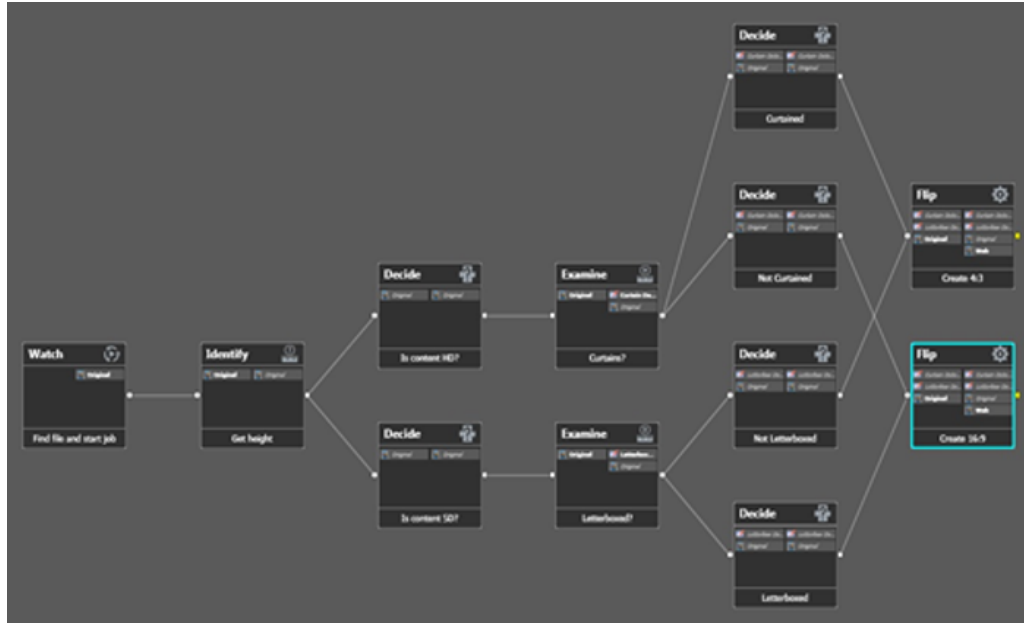
Configuring Transcoders

Create a new Vantage transcoder as described in previous section.

Set the *Working Directory* to either a local folder, or a path on a remote system. *You can only set a remote path for Vantage*. If you are setting a path to a remote system, a CIFS UNC path with the appropriate authentication credentials must be specified. The IP address specified in the UNC path must point to the remote computer running the Vantage SDK service.

Configuring Sources and Destinations

Use the following procedure to configure a source or destination for use with transcoders:

1. Open the DIVArchive Configuration Utility.

2. Navigate to the *Sources and Destinations* frame on the **System** tab.

3. Create a **LOCAL** Source/Destination for the Actor using the following parameters:

- *Source Name*: use the same name as the Actor name

- *IP Address*: leave this field empty

- *Source Type*: **LOCAL**

4. Configure the destination to include the following transcode options along with any other required *Connect Options*:

- **-tr_names** {TRANSCODER_NAME}

■ **-tr_restore_format** {WORKFLOW_NAME}

> **Note:** The auto format option is only valid for Telestream and BitScream.

For this example the *Connect Options* field is populated similar to the following:

```
-login diva -pass diva -tr_names vantage_001 -tr_restore_format TESTMINWorkflow
```

10

Integrating Movie2Me with DIVArchive

Movie2Me integration is available for use as a special Source/Destination. This Source/Destination uses a separate installation process and requires additional licensing. The integration is compatible with DIVArchive releases 7.2.1 and later. It performs similar to a combination of an FTP client and FTP server.

This chapter describes the general configuration and workflow of Movie2Me. Operational details can be found in the Movie2Me documentation delivered with your system. This chapter includes the following information:

- Minimum Requirements
- Movie2Me Workflow
- Configuring Movie2Me Integration

Minimum Requirements

The following are the minimum requirements for a complete Move2Me installation:

- DIVArchive Manager
- DIVArchive Actor
- Local Storage
- Remote Storage
- Local Disk Cache
- Two Movie2Me Cores
    - One Movie2Me Core local to the DIVArchive system
    - One Movie2Me Core remote from the DIVArchive system
- Cloud between the two cores

Movie2Me Workflow

The following figure and process described are representative of a basic Movie2Me workflow.
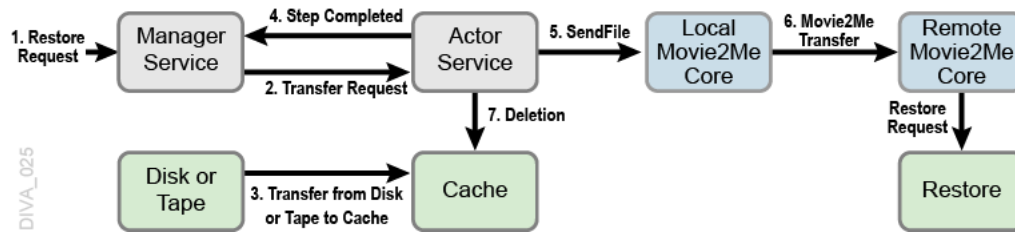
The step numbers in the following sequence refer to the numbering in the previous workflow diagram:

1. A Restore request is received by the Manager.

2. The Manager sends the transfer request to the selected Actor. The purpose of the transfer is to copy the content of the object to the Actor Cache (with or without AXF encapsulation).

3. The Actor performs the transfer from the disk or tape instance to its cache using a Source/Destination. The use of a Source/Destination permits more flexibility because you can specify some useful options including **-axf**, **-rm** and **-rxml**. The Actor reports the progress to the Manager as usual.

4. When the object is in the cache (transfer completed), the Manager receives a *Step Completed* message from the Actor indicating that the Actor will start the Movie2Me Transfer. This permits the Manager to release some resources (for example, the tape and the drive) used during the first transfer (to cache). The Manager resets the *Request Progress* to zero percent.

5. The Actor connects to a local Movie2Me Core and issues a `SendFile` command for each file being restored. This could actually be two files with an AXF Restore (`object.axf` and `Object.mdf`).

6. The local Movie2Me sends all files to the remote Movie2Me. While the transfer is running, the Actor monitors the progress and reports it to the Manager. When the transfer is one hundred percent complete the Actor sends a *Request Complete* message to the Manager.

7. The Manager sends a *Server Delete* request to the Actor, and the Actor deletes the temporary space in cache.

If you use Movie2Me during a Restore request to a system's DFM folder, and the request contains the **-rm** option, the `.mdf` file must be the last file transferred. Otherwise, an incomplete archive may result. If you use the **-axf** option instead, and the DFM folder is in *Single File Mode*, the order of the transfer does not matter.

The Movie2Me Core can be a restore folder, archive folder, DFM drop folder, and so on, depending on the request.

Configuring Movie2Me Integration

The first part of setting up the Movie2Me Integration requires that you configure a cache disk in DIVArchive, and then associate it with the Actor where the Movie2Me Local Core will be connecting.

An array and disk must be created. Only the name of the array and disk are relevant. Use the following procedure to setup the required array and disk:

1. Open the DIVArchive Configuration Utility.

2. Click the **Drives** tab, and then click the **+** button on the top of the *Arrays* frame to add an array.

3. Enter a name for the array (for example, `Move2MeArray`) in the *Array Name* field, and (optionally) add a description in the *Description* field.

4. Click **OK** to save the array.

5. Click the **Disk** tab, and then click the **+** button on the top of the *Disks* frame to add a disk. You can create multiple disks if there are multiple Movie2Me Cores in the system.

6. Enter a name for the disk in the *Disk Name* field, and select the array (for example, `Move2MeArray`) from the *Array* menu list.

7. Click **OK** to save the disk.

Next you must associate the disk with an Actor. Define the Actor that can see the new disk previously created in the Actor-Disk Connections Entry dialog box.

You must identify the *Mount Point* field as the root directory of the cache from the Actor's point of view.

For example, `cifs://Administrator:diva@\\172.16.3.201\Movie2MeWorkspace`.

The cache must be used for Movie2Me transfers *only*. Therefore, you must set the *Used For* field to **WAN_TRANSFER_CACHE_ONLY**.

The destination configuration defines how Actors will connect to the Local Movie2Me Core and the Peer ID of the Remote Movie2Me Core. The configuration requires the following *Connect Options*:

-login
The log in user name of the Local Movie2Me Core HTTP server.

-pass
The password associated with the user name of the Local Movie2Me Core HTTP server.

IP Address
The IP address of the Local Movie2Me Core.

-port
The listening port of the Local Movie2Me Core HTTP Server. In the following example the Actor uses the parameter to be able to connect to `http://172.16.3.201:8080/transferService`.

-peer
The ID of the Remote Movie2Me Core. You can obtain the ID from the Local Movie2Me Core as follows:

1. Stop the *m2m-core* service.

2. Open the file `C:\ProgramData\M2M-Core\database\m2mdb.script` file with a plain text editor (Notepad or Notepad++).

3. Search for entries similar to `INSERT INTO PEER VALUES(`.

4. The first value in braces is the *Peer-ID*.

The following limitations apply to this type of Source/Destination:

■ Archives are not supported from this type of Source/Destination.

■ N-Restore is not supported.

■ The *Cache Type* can be **Remote** or **Local** only (not FTP, Mediagrid, or Metasan).

- **Delete and Write** QOS is not supported.

- **Do Not Overwrite** QOS is not supported.

- VFR is not supported.

- This configuration is for a single Movie2Me Core for each DIVArchive Production System.

- Single Actor-Disk Connection with **WAN_TRANSFER** option per Actor. One Actor cannot see more than one **WAN_TRANSFER** cache.

After Movie2Me is installed, it uses its own utility for configuration as described below. The configuration should be the same on both cores as indicated.

First you must configure the local Movie2Me Core as follows:

1. Open the Movie2Me Core Configuration.

2. Enter the following information into the appropriate fields:

    Peer Name
    Enter the name of the local Movie2Me system.

    IP address
    The IP address must be set to either `localhost` or `127.0.0.1`.

    Port
    The local port number must be set to `9000`.

    Mode
    Set the Encryption Mode to `AES256` here and in the remote configuration.

    Passphrase
    Create a secure passphrase, enter it here. Remember this passphrase for the remote configuration.

    The following database settings refer to the local Movie2Me Database and not the DIVArchive Database. Confirm that the settings are correct.

    Type
    Use the menu list to select the correct type of database (typically `hsqldsb`).

    Path
    Enter the path to the database.

    Databasename
    Enter the name of the database.

    Username
    Enter the user name to connect to the database.

    Password
    Enter the password associated with the database connection user name.

    Workspace
    The Workspace is one of the most important settings because it identifies the workspace in the cache.

    The Workspace is the mount point identified in the DIVArchive configuration for the system cache. You create the mount point by setting up a disk in DIVArchive

and associating that disk with the selected Actor. It must be set to use the **WAN_Transfer_Cache_Only** *Usage Type*. This is a new *Usage Type*.

3. Click **Save** to save the configuration.

Next you must configure the Movie2Me Peer (remote) as follows:

1. Open the Movie2Me Peer Configuration.

2. Enter (at least) the following information into the appropriate fields:

   General Section
   Enter the appropriate information into each field in this section of the dialog box.

   Login Section
   Enter the appropriate information into each field in this section of the dialog box.

   Port (Connection Section)
   The port number must be the same as on the Local Movie2Me Configuration.

   Mode (Encryption Section)
   Select the appropriate option for the encryption mode so it is the same as the Local Movie2Me configuration. In this case select the *AES* option.

   Passphrase (Encryption Section)
   The encryption should be set the same as in the Local Movie2Me Configuration. Confirm that the Passphrase matches the entry in the local configuration.

3. Click **Save** to save the configuration.

11

Frequently Asked Questions

This chapter contains questions frequently asked during installation and configuration of the DIVArchive system and includes the answers to the following questions. Contact Oracle Support for any additional questions not covered here.

- What if the Customer Information Collection Tool does not work?
- Should all operating systems be kept up to date with critical updates?
- Should all operating systems be kept up to date with optional updates?
- Are there any operating system updates that should not be installed?
- Should the servers be restarted with any frequency?
- Should any services be restarted with any frequency?
- Should any vendor applications be restarted with any frequency?
- Should vendor applications always be updated to the latest version?
- What is the recommended frequency of database backups?
- Does Oracle recommend any particular database backup application?
- Where are the backup files located?
- Are there iterated versions of the database backup, and if so, how many are retained?
- Where are vendor-specific logs located?
- How far back in time do the logs go?
- What is the suggested log backup frequency?
- Are there any special considerations regarding maintenance and backup of vendor servers and systems?
- Are there any special considerations related to recovering from a server failure when the server is part of the vendor solution?

What if the Customer Information Collection Tool does not work?

Confirm that CygWin and the 7z archive programs are installed correctly. If the CygWin or the 7z programs are not installed, the Customer Information Collection Tool will stop running and display one of the following error messages:

```
Error: Cygwin environment could not be located at "...". Please check the
configuration or reinstall Cygwin environment if necessary.

Error: 7Z archiver could not be located at "...". Please check the configuration
```

```
or reinstall 7Z archiver if necessary.
```

Should all operating systems be kept up to date with critical updates?

Oracle recommends applying all critical updates to all computers.

Should all operating systems be kept up to date with optional updates?

Optional operating system updates are not necessary in the DIVArchive environment. However the decision to apply optional updates is left to your System Administrator.

Are there any operating system updates that should not be installed?

Oracle is not currently aware of any operating system updates that impact DIVArchive functionality or operations.

Should the servers be restarted with any frequency?

No, restarting the servers will cause downtime for the system and possibly cause data corruption if a process is executing when the server is restarted. Only restart a server when absolutely necessary and perform a normal system shutdown (do not just turn off the computer unless absolutely necessary).

Should any services be restarted with any frequency?

No, restarting the services will cause downtime for the system and possibly cause data corruption if a process is executing when the service is restarted. Only restart a service when absolutely necessary.

Should any vendor applications be restarted with any frequency?

No, only restart a vendor application when absolutely necessary.

Should vendor applications always be updated to the latest version?

No, only update vendor applications to benefit from new functionality or for bug fixes.

What is the recommended frequency of database backups?

The DIVArchive database automatically backs up every fifteen minutes.

Does Oracle recommend any particular database backup application?

A database backup service is provided in the DIVArchive package. You are welcome to use your own backup software as an additional security under the condition that you only backup the DIVArchive database backup files (in `H:\oraback`) and not the database itself.

Backing up the database directly is forbidden. For example, using Oracle RMAN or other non-DIVArchive database backup applications. Backing up the database directly with another program may interfere with the DIVArchive database backup service. This may render database restoration impossible using the embedded DIVArchive restore

utility, and could possibly result in data losses for which Oracle will accept no responsibility.

Where are the backup files located?

The database backup files are located on the Main Manager computer in the `H:\oraback` folder. The files are synced to the Backup Manager and an Actor in the `H:\oraback\mgr1` folder.

Are there iterated versions of the database backup, and if so, how many are retained?

The backup files are retained for the previous ten days. The retention period is configurable for the database backup files in the DIVArchive Backup Service configuration file. Contact Oracle Support for assistance.

Where are vendor-specific logs located?

The vendor-specific log files are located in the `%DIVA_HOME%\Program\log` folder in Windows and in the `/home/diva/DIVA/Program/log` directory in Linux.

How far back in time do the logs go?

The log file retention period is configurable in the DIVArchive configuration file. Contact Oracle Support for more information. The log files are retained as follows by default:

- Manager, DIVAnet: fifty hours

- Actor, Robot Manager, Storage Plan Manager, Avid Transfer Manager Communicator, Avid Archive Manager Communicator: ten days

- Drop Folder Monitor: variable based on size

What is the suggested log backup frequency?

The log files do not require backup.

Are there any special considerations regarding maintenance and backup of vendor servers and systems?

Oracle only supports the DIVArchive software. You must contact the server supplier for any hardware issue. You must keep Oracle in the loop for any issues on the DIVArchive solution (for example, loss of a RAID disk, failover to the backup manager, and so on).

Are there any special considerations related to recovering from a server failure when the server is part of the vendor solution?

As previously mentioned, you must keep Oracle in the loop if issues are encountered.

A

DIVArchive Options and Licensing

The following table identifies DIVArchive options and licensing metrics.

Part Number	Description	Licensing Metric
L101163	Oracle DIVArchive Nearline Capacity	Per TB
L101164	Oracle DIVArchive Archive Capacity	Per Slot
L101165	Oracle DIVArchive Actor	Per Server
L101166	Oracle DIVArchive Manager	Per Server
L101167	Oracle DIVArchive Partial File Restore	Per Wrapper
L101168	Oracle DIVArchive Avid Connectivity	Per Server
L101169	Oracle DIVArchive Application Filtering	Per Server
L101170	Oracle DIVArchive Storage Plan Manager (2 storage plans are included with a DIVArchive Manager License)	Per Server
L101171	Oracle DIVAnet	Per Server
L101172	Oracle DIVAdirector	Per User
L101918	Oracle DIVArchive Export / Import	Per Server
L101919	Oracle DIVArchive Additional Archive Robotic System	Per Tape Library
L101920	Oracle DIVArchive Automatic Data Migration	Per Server

B

Dynamic Configuration Changes

This appendix lists the currently supported changes to your configuration that become effective while the Manager is running, and those that require a software component or the DIVArchive Manager to be restarted. The following information is included:

- Updates in the Manager Configuration
- Updates in the Configuration Utility System Tab
  - Product Systems Frame
  - Sites Frame
  - Sources and Destinations Frame
  - Actors Frame
  - Transcoders Frame
- Updates in the Configuration Utility Robots Tab
  - Robot Managers Frame
  - Media Compatibility Frame
  - Robot Managers-ACS Frame
- Updates in the Configuration Utility Disks Tab
  - Arrays Frame
  - Disks Frame
  - Actor-Disk Connections Frame
- Updates in the Configuration Utility Drives Tab
  - Drives Frame
  - Libraries Frame
  - Drive Properties Frame
  - Actor-Drives Frame
- Updates in the Configuration Utility Tapes Tab
- Updates in the Configuration Utility Sets, Groups & Media Mapping Tab
- Updates in the Configuration Utility DIVAprotect Tab
- Updates in the Configuration Utility Storage Plans Tab
- Updates in the Configuration Utility Slots Tab
- Event Fields

- [Metrics Definitions](#)

- [Configuration Parameter Defaults and Values](#)

Updates in the Manager Configuration

If you change a parameter in the DIVArchive Manager configuration file the following list identifies what is currently required for your change to take effect.

You must use the `manager restart` command for the following parameter changes to take effect:

- `SERVICE_NAME` (also effective after reinstall)

- `DIVAMANAGER_NAME`

- `DIVAMANAGER_PORT`

- `DIVAMANAGER_TNSNAME`

- `DIVAMANAGER_DBHOST`

- `DIVAMANAGER_DBPORT`

- `DIVAMANAGER_DBSID`

- `DIVAMANAGER_DBUSER`

- `DIVAMANAGER_MAX_CONNECTIONS`

- `DIVAMANAGER_TYPICAL_OBJECT_SIZE`

- `DIVAMANAGER_CAPACITY_LOW_WATER_MARK`

- `DIVAMANAGER_STOP_IMMEDIATELY_FOR_REPACK`

- `DIVAMANAGER_TIME_TO_WAIT_FOR_GRACEFUL_SHUTDOWN`

- `DIVAMANAGER_DISMOUNT_AFTER`

- `DIVAMANAGER_UPDATE_PRIORITIES_PERIOD`

- `DIVAMANAGER_PING_INTERVAL`

- `DIVAMANAGER_ETC_FEATURE`

- `DIVAMANAGER_ETC_CONFIDENCE_LEVEL`

You must use the `manager reload` command for the following parameter changes to take effect:

- `DIVAMANAGER_TO_LOWER`

- `DIVAMANAGER_MAX_SIMULTANEOUS_REQUESTS`

- `DIVAMANAGER_MAX_INACTIVE_REQUESTS`

- `DIVAMANAGER_MAX_SPAN_SEGMENTS`

- `DIVAMANAGER_MAX_OBJECTS_FOR_REPACK`

- `DIVAMANAGER_MAX_DELAY_BETWEEN_SCHEDULER`

- `DIVAMANAGER_SCHEDULER_AFTER_INACTIVITY`

- `DIVAMANAGER_EXPORT_ROOT_DIR`

- `DIVAMANAGER_MAX_RESTORE_SERVERS`

- `DIVAMANAGER_MAX_EXPORT_TAPES`

- `DIVAMANAGER_MAX_EXPORT_ELEMENTS`

- `DIVAMANAGER_MAX_FILES_IN_ARCHIVE`

- `DIVAMANAGER_MAX_FILES_IN_PARTIAL_RESTORE`

- `USE_IMPROVED_BEST_WORST_FIT_ALGORITHM`

- `DIVAMANAGER_SITE_SUPPORT_ENABLED`

- `DIVAMANAGER_CACHE_QOS_USE_DISK`

- `DIVAMANAGER_PRIORITY_TIER`

- `DIVAMANAGER_OVERWRITE_POLICY`

- `DIVAMANAGER_OVERWRITE_OVERRIDE`

- `ATTEMPT_ACCESS_TO_OFFLINE_DISK`

- `CHANGE_DISK_STATE_ON_ERROR`

- `MANAGER_ACTOR_DISK_RETRY_NUMBER`

- `DISK_STATUS_POLLING_RATE`

- `DISK_BUFFER_SPACE`

- `DISK_CONNECTION_STATE_RESET_DELAY`

- `DIVAMANAGER_MAX_EXCLUDED_INSTANCES`

- `DIVAMANAGER_REQUEST_SCHEDULING_QUEUE_SIZE`

- `DIVAMANAGER_API_TASK_QUEUE_SIZE`

- `DIVAMANAGER_MAX_CONCURRENT_REQUESTS`

- `DIVAMANAGER_MIN_DB_CONNECTION_LIMIT`

- `DIVAMANAGER_MAX_DB_CONNECTION_LIMIT`

- `DIVAMANAGER_INITIAL_DB_CONNECTION_LIMIT`

- `DIVAMANAGER_INACTIVITY_TIMEOUT`

- `DIVAMANAGER_SIZE_OF_STATEMENT_CACHE`

- `DIVAMANAGER_DEFAULT_ROW_PREFETCH`

- `DIVAMANAGER_FAILOVER_ENABLED`

- `DIVAMANAGER_NUM_RS_SOLUTIONS_TO_EVALUATE`

- `DIVAMANAGER_DBSERVICENAME`

- `ABORT_ARCHIVES_ON_EMPTY_FILES` (reloadable in service mode)

- `TAPE_FULL_ON_SPAN_REJECTED` (reloadable in service mode)

Updates in the Configuration Utility System Tab

The following sections describe updates made in the various frames on the **System** tab.

Product Systems Frame

If one of the following parameters or actions in the *Production Systems* frame of the **Systems** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Add**

- **Delete**

- *Production System Name*

Sites Frame

If one of the following parameters or actions in the *Sites* frame of the **Systems** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Add**

- **Delete**

- *Site Name*

- *Is Main Site*

- *Comments*

Sources and Destinations Frame

If one of the following parameters or actions in the *Sources and Destinations* frame of the **Systems** tab is changed, you must **Notify Manager** for the changes to take effect.

Some changes only take effect after notifying the Manager, and currently executing requests are complete.

- **Add**

- **Delete** (**Notify Manager** and after requests complete)

- *Source Name* (**Notify Manager** and after requests complete)

- *IP Address* (**Notify Manager** and after requests complete)

- *Source Type* (**Notify Manager** and after requests complete)

- *Production System* (**Notify Manager** and after requests complete)

- *Site* (**Notify Manager** and after requests complete)

- *Connect Options* (**Notify Manager** and after requests complete)

- *Root Path* (**Notify Manager** and after requests complete)

- *Max Throughput* (**Notify Manager** and after requests complete)

- *Max Accesses* (**Notify Manager** and after requests complete). You must not make changes to this parameter while there are active request because it could lead to the request being terminated.

- *Max Read Accesses* (**Notify Manager** and after requests complete). You must not make changes to this parameter while there are active request because it could lead to the request being terminated.

- *Max Write Accesses* (**Notify Manager** and after requests complete). You must not make changes to this parameter while there are active request because it could lead to the request being terminated.

Actors Frame

If one of the following parameters or actions in the *Actors* frame of the **Systems** tab is changed, you must **Notify Manager** for the changes to take effect.

Before the change becomes effective on several of the parameters or actions, you must disconnect the Actor. Also, some changes only take effect after notifying the Manager, and currently executing requests are complete.

- **Add**

- **Delete** (must disconnect Actor first and **Notify Manager**)

- *Actor Name* (must disconnect Actor first and **Notify Manager**)

- *IP Address* (must disconnect Actor first and **Notify Manager**)

- *Port* (must disconnect Actor first and **Notify Manager**)

- *Production System* (**Notify Manager** and after requests complete)

- *Site* (**Notify Manager** and after requests complete)

- *Max Drive Operations* (**Notify Manager** and after requests complete)

- *Max Server Operations* (**Notify Manager** and after requests complete)

- *Max Disk Operations* (**Notify Manager** and after requests complete)

- *Direct Restore* (**Notify Manager** and after requests complete)

- *Cache Restore* (**Notify Manager** and after requests complete)

- *Copy to Group* (**Notify Manager** and after requests complete)

- *Associative Copy* (**Notify Manager** and after requests complete)

- *Repack* (**Notify Manager** and after requests complete)

- *Delete* (**Notify Manager** and after requests complete)

- *Direct Archive* (**Notify Manager** and after requests complete)

- *Cache Archive* (**Notify Manager** and after requests complete)

Transcoders Frame

If one of the following parameters or actions in the *Transcoders* frame of the **Systems** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Add**

- **Delete**

- *Transcoder Name*

- *Transcoder Type*

- *Transcoder Port*

- *Working Directory*

- *Executable Path*

- *Performance*

Updates in the Configuration Utility Robots Tab

The following sections describe updates made in the various frames on the **Robots** tab.

Robot Managers Frame

If one of the following parameters or actions in the *Robot Managers* frame of the **Robots** tab is changed, you must **Notify Manager** for the changes to take effect.

Before the change becomes effective on several of the parameters or actions, you must disconnect the Robot Manager.

- **Add**
- **Delete**
- *Robot Manager Name*
- *Address* (must disconnect Robot Manager first and **Notify Manager**)
- *Port* (must disconnect Robot Manager first and **Notify Manager**)
- *Site*

Media Compatibility Frame

If you **Delete** an entry in the *Media Compatibility* frame of the **Robots** tab, you must **Notify Manager** for the changes to take effect.

Robot Managers-ACS Frame

If you **Delete** an entry in the *Robot Managers-ACS* frame of the **Robots** tab, you must **Notify Manager** for the changes to take effect.

Updates in the Configuration Utility Disks Tab

The following sections describe updates made in the various frames on the **Disks** tab.

Arrays Frame

If one of the following parameters or actions in the *Arrays* frame of the **Disks** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Add**
- **Delete**
- *Array Name*
- *Description*

Disks Frame

If one of the following parameters or actions in the *Disks* frame of the **Disks** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Add**
- **Delete**
- *Disk Name*
- *Array*
- *Site*
- *Status*
- *Min Free Space*

Actor-Disk Connections Frame

If one of the following parameters or actions in the *Actor-Disk Connections* frame of the **Disks** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Add**
- **Delete**
- *Disk*
- *Actor*
- *Interface*
- *Mount Point*
- *Max Throughput*
- *Access*
- *Used For*

Updates in the Configuration Utility Drives Tab

The following sections describe updates made in the various frames on the **Drives** tab.

Drives Frame

If one of the following parameters or actions in the *Drives* frame of the **Drives** tab is changed, you must perform the noted action for the changes to take effect.

- **Delete** (**Notify Manager**)
- *Serial Number* (**Notify Manager**)
- *Status* (**Notify Manager**)
- *Enabled Operations* (**Notify Manager**)
- *Used* (`manager restart`)
- *Installation Date* (no action required, effective immediately)
- *Last Upgrade Date* (no action required, effective immediately)
- *Last Cleaning Date* (no action required, effective immediately)

Libraries Frame

If one of the following parameters or actions in the *Libraries* frame of the **Drives** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Delete**
- *Name*
- *Serial Number*
- *Status*

Drive Properties Frame

If one of the following parameters or actions in the *Drive Properties* frame of the **Drives** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Add** (through `syncDB`)

- **Delete**

Actor-Drives Frame

If one of the following parameters or actions in the *Actor-Drives* frame of the **Drives** tab is changed, you must **Notify Manager** for the changes to take effect.

- **Add**
- **Delete**
- *Actor*
- *Drive*

Updates in the Configuration Utility Tapes Tab

If one of the following parameters or actions in the **Tapes** tab is changed, you must perform the noted action for the changes to take effect.

- *Tape Properties* (**Notify Manager**)
- *Empty Ejected Tapes* (no action required, effective immediately)
- *Inserted Protected Tapes* (no action required, effective immediately)
- *Tape States* (no action required, effective immediately)

Updates in the Configuration Utility Sets, Groups & Media Mapping Tab

Changes made in this tab are effective as soon as they are applied. No manual update is necessary.

Updates in the Configuration Utility DIVAprotect Tab

If one of the following parameters or actions in the **DIVAprotect** tab is changed, you must perform the noted action for the changes to take effect.

- Configuration (**Notify Manager**)
- Event Definitions (currently cannot be altered)
- Metric Definitions (no action required, effective immediately)

Updates in the Configuration Utility Storage Plans Tab

Changes made in this tab are effective immediately. *It is highly recommended that the Storage Plan Manager Service be stopped before altering any setting in this tab*.

Updates in the Configuration Utility Slots Tab

Changes made in this tab are effective immediately. *It is highly recommended that the Storage Plan Manager Service be stopped before altering any setting in this tab*.

Event Fields

The following three tables identify event fields and the types of events associated with them. *There are three tables only due to the amount of entries*. Locate the desired field on

the top row of the table, and then follow down the column to identify which events are valid for the selected field.

Table B–1   Event Fields Table One

	Event Type	Tape Type	Tape Barcode	Drive Type	Drive Name	Disk Name	Drive Serial Number	Library Serial Number	SD Name	Actor Name
TAPE_INSERT	Yes	Yes	Yes					Yes		
TAPE_INSERT_ERR	Yes							Yes		
TAPE_MOUNT	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_MOUNT_ERR	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_POSITION	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_POSITION_ERR	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_READ	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_READ_ERR	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_WRITE	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_WRITE_ERR	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
DISK_READ[1]	Yes					Yes				Yes
DISK_READ_ERR[1]	Yes					Yes				Yes
DISK_WRITE[1]	Yes					Yes				Yes
DISK_WRITE_ERR[1]	Yes					Yes				Yes
SD_READ	Yes								Yes	Yes
SD_READ_ERR	Yes								Yes	Yes
SD_WRITE	Yes								Yes	Yes
SD_WRITE_ERR	Yes								Yes	Yes
TAPE_UNLOAD	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_UNLOAD_ERR	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_DISMOUNT	Yes	Yes	Yes	Yes	Yes		Yes	Yes		
TAPE_DISMOUNT_ERR	Yes	Yes	Yes	Yes	Yes		Yes	Yes		
TAPE_EJECT	Yes	Yes	Yes					Yes		
TAPE_EJECT_ERR	Yes	Yes	Yes					Yes		
END_OF_TAPE	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
TAPE_REPACK	Yes							Yes		
ARCHIVE_REQUEST	Yes								Yes	
COPY_REQUEST	Yes									
COPY_AS_REQUEST (to new)	Yes									
CREATE_INSTANCE	Yes									
RESTORE and PARTIAL_RESTORE	Yes								Yes	
DELETE_OBJECT	Yes									
DELETE_INSTANCE	Yes									
TRANSCODE_END	Yes									Yes
TRANSCODE_ERR	Yes									Yes
STOPPED_ON_CANCEL	Yes									

Table B–1   (Cont.) Event Fields Table One

	Event Type	Tape Type	Tape Barcode	Drive Type	Drive Name	Disk Name	Drive Serial Number	Library Serial Number	SD Name	Actor Name
CHECKSUM_ERROR_TAPE	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
CHECKSUM_ERROR_DISK	Yes					Yes				Yes
CHECKSUM_ERROR_SD	Yes								Yes	Yes
TAPE_IMPORT	Yes		Yes							
TAPE_EXPORT	Yes		Yes							

[1]   The transcoder work directory is not a DIVArchive disk. No **DISK READ** or **DISK WRITE** events are created when accessing this directory.

The presence of *Optional* in the following table indicates that it is optional. New Instance IDs are only generated after the final write to the destination media. Instance ID is not available in the following cases:

■   Temporary instances created in cache disk by an Archive request

■   SD READ or SD WRITE during the transcode phase of an archive when transferring to or from the transcoder work directory

■   Cache DISK READ or DISK WRITE when performing a tape to tape Copy request

■   Tape positioning before a tape write (Archive request)

■   End Of Tape (EOT exception) encountered during an Archive request

Table B–2   Event Fields Table Two

	Object Name[1]	Object Category[1]	Object Instance[1]	Media (group or array)	Request ID	Event End Time	Event Duration	Transfer Size	Transfer Rate
TAPE_INSERT						Yes	Yes		
TAPE_INSERT_ERR				Yes		Yes			
TAPE_MOUNT				Yes		Yes	Yes		
TAPE_MOUNT_ERR				Yes		Yes			
TAPE_POSITION	Yes	Yes	Optional	Yes	Yes	Yes	Yes		
TAPE_POSITION_ERR	Yes	Yes	Optional	Yes	Yes	Yes			
TAPE_READ	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TAPE_READ_ERR	Yes	Yes	Yes	Yes	Yes	Yes		Yes	
TAPE_WRITE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TAPE_WRITE_ERR	Yes	Yes		Yes	Yes	Yes		Yes	
DISK_READ[2]	Yes	Yes	Optional	Yes	Yes	Yes	Yes	Yes	Yes
DISK_READ_ERR[2]	Yes	Yes	Optional	Yes	Yes	Yes		Yes	
DISK_WRITE[2]	Yes	Yes	Optional	Yes	Yes	Yes	Yes	Yes	Yes
DISK_WRITE_ERR[2]	Yes	Yes		Yes	Yes	Yes		Yes	
SD_READ	Yes	Yes	Optional		Yes	Yes	Yes	Yes	Yes
SD_READ_ERR	Yes	Yes	Optional		Yes	Yes		Yes	
SD_WRITE	Yes	Yes	Optional		Yes	Yes	Yes	Yes	Yes
SD_WRITE_ERR	Yes	Yes			Yes	Yes		Yes	
TAPE_UNLOAD				Yes		Yes	Yes		
TAPE_UNLOAD_ERR				Yes		Yes			

Table B–2 (Cont.) Event Fields Table Two

	Object Name[1]	Object Category[1]	Object Instance[1]	Media (group or array)	Request ID	Event End Time	Event Duration	Transfer Size	Transfer Rate
TAPE_DISMOUNT				Yes		Yes	Yes		
TAPE_DISMOUNT_ERR				Yes		Yes			
TAPE_EJECT						Yes	Yes		
TAPE_EJECT_ERR						Yes			
END_OF_TAPE	Yes	Yes	Optional	Yes	Yes	Yes			
TAPE_REPACK					Yes	Yes			
ARCHIVE_REQUEST	Yes	Yes		Yes	Yes	Yes	Yes	Yes	
COPY_REQUEST	Yes	Yes		Yes	Yes	Yes	Yes	Yes	
COPY_AS_REQUEST (to new)	Yes	Yes		Yes	Yes	Yes	Yes	Yes	
CREATE_INSTANCE	Yes		Yes	Yes	Yes	Yes		Yes	
RESTORE and PARTIAL_RESTORE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
DELETE_OBJECT	Yes	Yes			Yes	Yes			
DELETE_INSTANCE	Yes	Yes	Yes	Yes	Yes	Yes		Yes	
TRANSCODE_END	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes
TRANSCODE_ERR	Yes	Yes	Yes		Yes	Yes			
STOPPED_ON_CANCEL	Yes	Yes			Yes	Yes			
CHECKSUM_ERROR_TAPE	Yes	Yes	Optional	Yes	Yes	Yes			
CHECKSUM_ERROR_DISK	Yes	Yes	Optional	Yes	Yes	Yes			
CHECKSUM_ERROR_SD	Yes	Yes	Optional		Yes	Yes			
TAPE_IMPORT				Yes		Yes			
TAPE_EXPORT				Yes	Yes	Yes			

[1] Object information is not provided for Repack requests.
[2] The transcoder work directory is not a DIVArchive disk. No **DISK READ** or **DISK WRITE** events are created when accessing this directory.

Table B–3 Event Fields Table Three

	Transfer Error Rate	Error Code	Error Message	Transcoder or Analyzer Name	Number of Archive Operations	Data Size
TAPE_INSERT						
TAPE_INSERT_ERR		Yes	Yes			
TAPE_MOUNT						
TAPE_MOUNT_ERR		Yes	Yes			
TAPE_POSITION						
TAPE_POSITION_ERR		Yes	Yes			
TAPE_READ	Yes					
TAPE_READ_ERR		Yes	Yes			
TAPE_WRITE	Yes					
TAPE_WRITE_ERR		Yes	Yes			
DISK_READ[1]						

Table B–3   (Cont.) Event Fields Table Three

	Transfer Error Rate	Error Code	Error Message	Transcoder or Analyzer Name	Number of Archive Operations	Data Size
DISK_READ_ERR[1]		Yes	Yes			
DISK_WRITE[1]						
DISK_WRITE_ERR[1]		Yes	Yes			
SD_READ						
SD_READ_ERR		Yes	Yes			
SD_WRITE						
SD_WRITE_ERR		Yes	Yes			
TAPE_UNLOAD						
TAPE_UNLOAD_ERR		Yes	Yes			
TAPE_DISMOUNT						
TAPE_DISMOUNT_ERR		Yes	Yes			
TAPE_EJECT						
TAPE_EJECT_ERR		Yes	Yes			
END_OF_TAPE						
TAPE_REPACK						
ARCHIVE_REQUEST					Yes	
COPY_REQUEST					Yes	
COPY_AS_REQUEST (to new)					Yes	
CREATE_INSTANCE						
RESTORE and PARTIAL_RESTORE					Yes	
DELETE_OBJECT						
DELETE_INSTANCE						
TRANSCODE_END				Yes		
TRANSCODE_ERR		Yes	Yes	Yes		
STOPPED_ON_CANCEL						
CHECKSUM_ERROR_TAPE						
CHECKSUM_ERROR_DISK						
CHECKSUM_ERROR_SD						
TAPE_IMPORT						Yes
TAPE_EXPORT						Yes

[1]   The transcoder work directory is not a DIVArchive disk. No **DISK READ** or **DISK WRITE** events are created when accessing this directory.

Metrics Definitions

The following table identifies DIVArchive metrics definitions. By default, all definitions are enabled.

Table B–4   Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
ACTOR_READ_WRITE	Actor: amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Actor Name	Lifetime
ACTOR_READ_WRITE_ ABORTED_NUMBER	Actor: number of terminated READ and terminated WRITE operations with drives	TAPE_READ_ERR TAPE_WRITE_ERR	Count	Null	Event ID	Actor Name	Lifetime
ACTOR_READ_WRITE_ ABORTED_NUMBER_ DAY	Actor: number of terminated READ and terminated WRITE operations with drives	TAPE_READ_ERR TAPE_WRITE_ERR	Count	Null	Event ID	Actor Name	Day
ACTOR_READ_WRITE_ ABORTED_NUMBER_SD	Actor: number of terminated READ and terminated WRITE operations with SD	SD_READ_ERR SD_WRITE_ERR	Count	Null	Event ID	Actor Name	Lifetime
ACTOR_READ_WRITE_ ABORTED_NUMBER_ SD_DAY	Actor: number of terminated READ and terminated WRITE operations with SD	SD_READ_ERR SD_WRITE_ERR	Count	Null	Event ID	Actor Name	Day
ACTOR_READ_WRITE_ DAY	Actor: amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Actor Name	Day
ACTOR_READ_WRITE_ MONTH	Actor: amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Actor Name	Month
ACTOR_READ_WRITE_ NUMBER	Actor: number of READ and WRITE operations	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Actor Name	Lifetime

Metrics Definitions

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
ACTOR_READ_WRITE_NUMBER_DAY	Actor: number of READ and WRITE operations	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Actor Name	Day
ACTOR_READ_WRITE_NUMBER_MONTH	Actor: number of READ and WRITE operations	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Actor Name	Month
ACTOR_TIME_ALL_OPERATION	Actor: time in all operations	DISK_READ DISK_READ_ERR DISK_WRITE DISK_WRITE_ERR SD_READ SD_READ_ERR SD_WRITE SD_WRITE_ERR TAPE_END_OF_TAPE TAPE_MOUNT_ERR TAPE_POSITION TAPE_POSITION_ERR TAPE_READ TAPE_READ_ERR TAPE_UNLOAD TAPE_UNLOAD_ERR TAPE_WRITE TAPE_WRITE_ERR	Sum	Null	Duration	Actor Name	Lifetime

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
ACTOR_TIME_ALL_ OPERATION_DAY	Actor: time in all operations	DISK_READ DISK_READ_ERR DISK_WRITE DISK_WRITE_ERR SD_READ SD_READ_ERR SD_WRITE SD_WRITE_ERR TAPE_END_OF_TAPE TAPE_MOUNT_ERR TAPE_POSITION TAPE_POSITION_ERR TAPE_READ TAPE_READ_ERR TAPE_UNLOAD TAPE_UNLOAD_ERR TAPE_WRITE TAPE_WRITE_ERR	Sum	Null	Duration	Actor Name	Day
ACTOR_TIME_ALL_ OPERATION_MONTH	Actor: time in all operations	DISK_READ DISK_READ_ERR DISK_WRITE DISK_WRITE_ERR SD_READ SD_READ_ERR SD_WRITE SD_WRITE_ERR TAPE_END_OF_TAPE TAPE_MOUNT_ERR TAPE_POSITION TAPE_POSITION_ERR TAPE_READ TAPE_READ_ERR TAPE_UNLOAD TAPE_UNLOAD_ERR TAPE_WRITE TAPE_WRITE_ERR	Sum	Null	Duration	Actor Name	Month
ACTOR_TIME_READ	Actor: time in READ operations	DISK_READ SD_READ TAPE_READ	Sum	Null	Duration	Actor Name	Lifetime
ACTOR_TIME_READ_ DAY	Actor: time in READ operations	DISK_READ SD_READ TAPE_READ	Sum	Null	Duration	Actor Name	Day
ACTOR_TIME_READ_ MONTH	Actor: time in READ operations	DISK_READ SD_READ TAPE_READ	Sum	Null	Duration	Actor Name	Month

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
ACTOR_TIME_WRITE	Actor: time in WRITE operations	DISK_WRITE SD_WRITE TAPE_WRITE	Sum	Null	Duration	Actor Name	Lifetime
ACTOR_TIME_WRITE_ DAY	Actor: time in WRITE operations	DISK_WRITE SD_WRITE TAPE_WRITE	Sum	Null	Duration	Actor Name	Day
ACTOR_TIME_WRITE_ MONTH	Actor: time in WRITE operations	DISK_WRITE SD_WRITE TAPE_WRITE	Sum	Null	Duration	Actor Name	Month
DISK_AVG_ TRANSFER_RATE_ READ	DISK: average transfer rate of READ	DISK_READ	Average	Null	Transfer Rate	Disk Name	Lifetime
DISK_AVG_ TRANSFER_RATE_ READ_DAY	DISK: average transfer rate of READ	DISK_READ	Average	Null	Transfer Rate	Disk Name	Day
DISK_AVG_ TRANSFER_RATE_ READ_MONTH	DISK: average transfer rate of READ	DISK_READ	Average	Null	Transfer Rate	Disk Name	Month
DISK_AVG_ TRANSFER_RATE_ WRITE	DISK: average transfer rate of WRITE	DISK_WRITE	Average	Null	Transfer Rate	Disk Name	Lifetime
DISK_AVG_ TRANSFER_RATE_ WRITE_DAY	DISK: average transfer rate of WRITE	DISK_WRITE	Average	Null	Transfer Rate	Disk Name	Day
DISK_AVG_ TRANSFER_RATE_ WRITE_MONTH	DISK: average transfer rate of WRITE	DISK_WRITE	Average	Null	Transfer Rate	Disk Name	Month
DISK_CHECKSUM_ FAILURE_COUNT_DAY	DISK: Checksum failure operations count	CHECKSUM_ERROR_ DISK	Count	Null	Event ID	Disk Name	Day
DISK_CHECKSUM_ FAILURE_COUNT_ MONTH	DISK: Checksum Failure Operations Count	CHECKSUM_ERROR_ DISK	Count	Null	Event ID	Disk Name	Month
DISK_NUMBER_READ	Disk: Total number of READ operations	DISK_READ DISK_READ_ERR	Count	Null	Event ID	Disk Name	Lifetime
DISK_NUMBER_READ_ ABORTED	Disk: Total number of terminated READ operations	DISK_READ_ERR	Count	Null	Event ID	Disk Name	Lifetime
DISK_NUMBER_READ_ ABORTED_DAY	Disk: Total number of terminated READ operations	DISK_READ_ERR	Count	Null	Event ID	Disk Name	Day
DISK_NUMBER_READ_ ABORTED_MONTH	Disk: Total number of terminated READ operations	DISK_READ_ERR	Count	Null	Event ID	Disk Name	Month

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
DISK_NUMBER_READ_ DAY	Disk: Total number of READ operations	DISK_READ DISK_READ_ERR	Count	Null	Event ID	Disk Name	Day
DISK_NUMBER_READ_ MONTH	Disk: Total number of READ operations	DISK_READ DISK_READ_ERR	Count	Null	Event ID	Disk Name	Month
DISK_NUMBER_WRITE	Disk: Total number of WRITE operations	DISK_WRITE DISK_WRITE_ERR	Count	Null	Event ID	Disk Name	Lifetime
DISK_NUMBER_ WRITE_ABORTED	Disk: Total number of terminated WRITE operations	DISK_WRITE_ERR	Count	Null	Event ID	Disk Name	Lifetime
DISK_NUMBER_ WRITE_ABORTED_DAY	Disk: Total number of terminated WRITE operations	DISK_WRITE_ERR	Count	Null	Event ID	Disk Name	Day
DISK_NUMBER_ WRITE_ABORTED_ MONTH	Disk: Total number of terminated WRITE operations	DISK_WRITE_ERR	Count	Null	Event ID	Disk Name	Month
DISK_NUMBER_ WRITE_DAY	Disk: Total number of WRITE operations	DISK_WRITE DISK_WRITE_ERR	Count	Null	Event ID	Disk Name	Day
DISK_NUMBER_ WRITE_MONTH	Disk: Total number of WRITE operations	DISK_WRITE DISK_WRITE_ERR	Count	Null	Event ID	Disk Name	Month
DISK_READ	DISK: total amount of data READ	DISK_READ	Sum	Null	Transfer Size	Disk Name	Lifetime
DISK_READ_DAY	DISK: total amount of data READ	DISK_READ	Sum	Null	Transfer Size	Disk Name	Day
DISK_READ_MONTH	DISK: total amount of data READ	DISK_READ	Sum	Null	Transfer Size	Disk Name	Month
DISK_TIME_ALL_ OPERATION	DISK: total time of ALL operations	DISK_READ DISK_WRITE	Sum	Null	Duration	Disk Name	Lifetime
DISK_TIME_ALL_ OPERATION_DAY	DISK: total time of ALL operations	DISK_READ DISK_WRITE	Sum	Null	Duration	Disk Name	Day
DISK_TIME_ALL_ OPERATION_MONTH	DISK: total time of ALL operations	DISK_READ DISK_WRITE	Sum	Null	Duration	Disk Name	Month
DISK_TIME_READ	DISK: total time of READ operations	DISK_READ	Sum	Null	Duration	Disk Name	Lifetime
DISK_TIME_READ_ DAY	DISK: total time of READ operations	DISK_READ	Sum	Null	Duration	Disk Name	Day

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
DISK_TIME_READ_MONTH	DISK: total time of READ operations	DISK_READ	Sum	Null	Duration	Disk Name	Month
DISK_TIME_WRITE	DISK: total time of WRITE operations	DISK_WRITE	Sum	Null	Duration	Disk Name	Lifetime
DISK_TIME_WRITE_DAY	DISK: total time of WRITE operations	DISK_WRITE	Sum	Null	Duration	Disk Name	Day
DISK_TIME_WRITE_MONTH	DISK: total time of WRITE operations	DISK_WRITE	Sum	Null	Duration	Disk Name	Month
DISK_WRITE	DISK: total amount of data WRITE	DISK_WRITE	Sum	Null	Transfer Size	Disk Name	Lifetime
DISK_WRITE_DAY	DISK: total amount of data WRITE	DISK_WRITE	Sum	Null	Transfer Size	Disk Name	Day
DISK_WRITE_MONTH	DISK: total amount of data WRITE	DISK_WRITE	Sum	Null	Transfer Size	Disk Name	Month
DIVARCHIVE_SYSTEM_ACTIVE_ARCHIVE_NUMBER	DIVArchive System: number of active Archive requests	ARCHIVE_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Lifetime
DIVARCHIVE_SYSTEM_ACTIVE_ARCHIVE_NUMBER_DAY	DIVArchive System: number of active Archive requests	ARCHIVE_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Day
DIVARCHIVE_SYSTEM_ACTIVE_ARCHIVE_NUMBER_MONTH	DIVArchive System: number of active Archive requests	ARCHIVE_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Month
DIVARCHIVE_SYSTEM_ACTIVE_COPY_AS_NUMBER	DIVArchive System: number of active Copy As New Object requests	COPY_AS_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Lifetime
DIVARCHIVE_SYSTEM_ACTIVE_COPY_AS_NUMBER_DAY	DIVArchive System: number of active Copy As New Object requests	COPY_AS_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Day
DIVARCHIVE_SYSTEM_ACTIVE_COPY_AS_NUMBER_MONTH	DIVArchive System: number of active Copy As New Object requests	COPY_AS_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Month

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
DIVARCHIVE_ SYSTEM_ACTIVE_ COPY_NUMBER	DIVArchive System: number of active Copy requests	COPY_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_ACTIVE_ COPY_NUMBER_DAY	DIVArchive System: number of active Copy requests	COPY_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_ACTIVE_ COPY_NUMBER_MONTH	DIVArchive System: number of active Copy requests	COPY_REQUEST	Maximum	Null	Number of Operations	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_ACTIVE_ RESTORE_NUMBER	DIVArchive System: number of active Restore requests	RESTORE	Maximum	Null	Number of Operations	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_ACTIVE_ RESTORE_NUMBER_ DAY	DIVArchive System: number of active Restore requests	RESTORE	Maximum	Null	Number of Operations	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_ACTIVE_ RESTORE_NUMBER_ MONTH	DIVArchive System: number of active Restore requests	RESTORE	Maximum	Null	Number of Operations	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_AVG_READ_ WRITE	DIVArchive System: average amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Weighted Average	Duration	Transfer Size	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_AVG_READ_ WRITE_DAY	DIVArchive System: average amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Weighted Average	Duration	Transfer Size	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_AVG_READ_ WRITE_MONTH	DIVArchive System: average amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Weighted Average	Null	Transfer Size	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_ARCHIVE	DIVArchive System: number of objects archived	ARCHIVE_REQUEST	Count	Null	Transfer Size	Local DIVArchive System	Lifetime

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_ARCHIVE_ DAY	DIVArchive System: number of objects archived	ARCHIVE_REQUEST	Count	Null	Transfer Size	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_ARCHIVE_ MONTH	DIVArchive System: number of objects archived	ARCHIVE_REQUEST	Count	Null	Event ID	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_CREATED	DIVArchive System: number of objects created	ARCHIVE_REQUEST COPY_AS_REQUEST	Count	Null	Event ID	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_CREATED_ DAY	DIVArchive System: number of objects created	ARCHIVE_REQUEST COPY_AS_REQUEST	Count	Null	Event ID	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_CREATED_ MONTH	DIVArchive System: number of objects created	ARCHIVE_REQUEST COPY_AS_REQUEST	Count	Null	Event ID	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_DELETED	DIVArchive System: number of objects deleted	DELETE_OBJECT	Count	Null	Event ID	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_DELETED_ DAY	DIVArchive System: number of objects deleted	DELETE_OBJECT	Count	Null	Event ID	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_DELETED_ MONTH	DIVArchive System: number of objects deleted	DELETE_OBJECT	Count	Null	Event ID	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ COPY	DIVArchive System: number of object instances copied	COPY_REQUEST	Count	Null	Event ID	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ COPY_DAY	DIVArchive System: number of object instances copied	COPY_REQUEST	Count	Null	Event ID	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ COPY_MONTH	DIVArchive System: number of object instances copied	COPY_REQUEST	Count	Null	Event ID	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ CREATED	DIVArchive System: number of object instances created	CREATE_INSTANCE	Count	Null	Event ID	Local DIVArchive System	Lifetime

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ CREATED_DAY	DIVArchive System: number of object instances created	CREATE_INSTANCE	Count	Null	Event ID	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ CREATED_MONTH	DIVArchive System: number of object instances created	CREATE_INSTANCE	Count	Null	Event ID	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ DELETED	DIVArchive System: number of object instances deleted	DELETE_INSTANCE	Count	Null	Event ID	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ DELETED_DAY	DIVArchive System: number of object instances deleted	DELETE_INSTANCE	Count	Null	Event ID	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_INSTANCE_ DELETED_MONTH	DIVArchive System: number of object instances deleted	DELETE_INSTANCE	Count	Null	Event ID	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_RESTORE	DIVArchive System: number of objects restored	RESTORE	Count	Null	Event ID	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_RESTORE_ DAY	DIVArchive System: number of objects restored	RESTORE	Count	Null	Event ID	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_NUMBER_ OBJECT_RESTORE_ MONTH	DIVArchive System: number of objects restored	RESTORE	Count	Null	Event ID	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_READ_WRITE	DIVArchive System: amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_READ_ WRITE_ABORTED_ NUMBER	DIVArchive System: number of terminated READ and terminated WRITE operations	DISK_READ_ERR DISK_WRITE_ERR SD_READ_ERR SD_WRITE_ERR TAPE_READ_ERR TAPE_WRITE_ERR	Count	Null	Event ID	Local DIVArchive System	Lifetime

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
DIVARCHIVE_ SYSTEM_READ_ WRITE_ABORTED_ NUMBER_DAY	DIVArchive System: number of terminated READ and terminated WRITE operations	DISK_READ_ERR DISK_WRITE_ERR SD_READ_ERR SD_WRITE_ERR TAPE_READ_ERR TAPE_WRITE_ERR	Count	Null	Event ID	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_READ_ WRITE_ABORTED_ NUMBER_MONTH	DIVArchive System: number of terminated READ and terminated WRITE operations	DISK_READ_ERR DISK_WRITE_ERR SD_READ_ERR SD_WRITE_ERR TAPE_READ_ERR TAPE_WRITE_ERR	Count	Null	Event ID	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_READ_ WRITE_DAY	DIVArchive System: amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_READ_ WRITE_MONTH	DIVArchive System: amount of data READ and written	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Local DIVArchive System	Month
DIVARCHIVE_ SYSTEM_READ_ WRITE_NUMBER	DIVArchive System: number of READ and WRITE operations	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Local DIVArchive System	Lifetime
DIVARCHIVE_ SYSTEM_READ_ WRITE_NUMBER_DAY	DIVArchive System: number of READ and WRITE operations	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Local DIVArchive System	Day
DIVARCHIVE_ SYSTEM_READ_ WRITE_NUMBER_ MONTH	DIVArchive System: number of READ and WRITE operations	DISK_READ DISK_WRITE SD_READ SD_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Local DIVArchive System	Month
MEDIA_ARCHIVED_ OBJECT_DATASIZE_ DAY	Media: data size of all objects archived	ARCHIVE_REQUEST	Sum	Null	Transfer Size	Media Name	Day

Table B–4 (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
MEDIA_ARCHIVED_ OBJECT_DATASIZE_ MONTH	Media: data size of all objects archived	ARCHIVE_REQUEST	Sum	Null	Transfer Size	Media Name	Month
MEDIA_OBJECT_ INSTANCE_CREATE	Media: number of object instances created	CREATE_INSTANCE	Count	Null	Event ID	Media Name	Lifetime
MEDIA_OBJECT_ INSTANCE_CREATE_ DAY	Media: number of object instances created	CREATE_INSTANCE	Count	Null	Event ID	Media Name	Day
MEDIA_OBJECT_ INSTANCE_CREATE_ MONTH	Media: number of object instances created and deleted	CREATE_INSTANCE	Count	Null	Event ID	Media Name	Month
MEDIA_OBJECT_ INSTANCE_DELETE	Media: number of object instances deleted	DELETE_INSTANCE	Count	Null	Event ID	Media Name	Lifetime
MEDIA_OBJECT_ INSTANCE_DELETE_ DAY	Media: number of object instances deleted	DELETE_INSTANCE	Count	Null	Event ID	Media Name	Day
MEDIA_OBJECT_ INSTANCE_DELETE_ MONTH	Media: number of object instances created and deleted	DELETE_INSTANCE	Count	Null	Event ID	Media Name	MOnth
MEDIA_READ_WRITE	Media: amount of data READ and written	DISK_READ DISK_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Media Name	Lifetime
MEDIA_READ_WRITE_ DAY	Media: amount of data READ and written	DISK_READ DISK_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Media Name	Day
MEDIA_READ_WRITE_ MONTH	Media: amount of data READ and written	DISK_READ DISK_WRITE TAPE_READ TAPE_WRITE	Sum	Null	Transfer Size	Media Name	Month
MEDIA_READ_WRITE_ NUMBER	Media: number of READ and WRITE operations	DISK_READ DISK_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Media Name	Lifetime

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
MEDIA_READ_WRITE_ NUMBER_DAY	Media: number of READ and WRITE operations	DISK_READ DISK_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Media Name	Day
MEDIA_READ_WRITE_ NUMBER_MONTH	Media: number of READ and WRITE operations	DISK_READ DISK_WRITE TAPE_READ TAPE_WRITE	Count	Null	Event ID	Media Name	Month
MEDIA_RESTORE_ OBJECT_DATASIZE_ DAY	Media: data size of all objects restored	RESTORE	Sum	Null	Transfer Size	Media Name	Day
MEDIA_RESTORE_ OBJECT_DATASIZE_ MONTH	Media: data size of all objects restored	RESTORE	Sum	Null	Transfer Size	Media Name	Month
MEDIA_TAPE_ EXPORT_NUMBER_DAY	Media: Number of tapes EXPORTED	TAPE_EXPORT	Count	Null	Event ID	Media Name	Day
MEDIA_TAPE_ EXPORT_NUMBER_ MONTH	Media: Number of tapes EXPORTED	TAPE_EXPORT	Count	Null	Event ID	Media Name	Month
MEDIA_TAPE_ IMPORT_NUMBER_DAY	Media: Number of tapes IMPORTED	TAPE_IMPORT	Count	Null	Event ID	Media Name	Day
MEDIA_TAPE_ IMPORT_NUMBER_ MONTH	Media: Number of tapes IMPORTED	TAPE_IMPORT	Count	Null	Event ID	Media Name	Month
SD_ARCHIVE_ OBJECT_DATASIZE_ DAY	SD: data size of all objects archived	ARCHIVE_REQUEST	Sum	Null	Transfer Size	SD Name	Day
SD_ARCHIVE_ OBJECT_DATASIZE_ MONTH	SD: data size of all objects archived	ARCHIVE_REQUEST	Sum	Null	Transfer Size	SD Name	Month
SD_CHECKSUM_ FAILURE_COUNT_DAY	SD: checksum failure operations count	CHECKSUM_ERROR_SD	Count	Null	Event ID	SD Name	Day
SD_READ	SD: amount of data READ	SD_READ	Sum	Null	Transfer ID	SD Name	Lifetime
SD_READ_DAY	SD: amount of data READ	SD_READ	Sum	Null	Transfer ID	SD Name	Day
SD_READ_MONTH	SD: amount of data READ	SD_READ	Sum	Null	Transfer Id	SD Name	Month
SD_READ_NUMBER	SD: number of READ operations	SD_READ	Count	Null	Event ID	SD Name	Lifetime
SD_READ_NUMBER_ DAY	SD: number of READ operations	SD_READ	Count	Null	Event ID	SD Name	Day

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
SD_READ_NUMBER_ MONTH	SD: number of READ operations	SD_READ	Count	Null	Event ID	SD Name	Month
SD_RESTORE_ OBJECT_DATASIZE_ DAY	SD: data size of all objects restored	RESTORE	Sum	Null	Transfer Size	SD Name	Day
SD_RESTORE_ OBJECT_DATASIZE_ MONTH	SD: data size of all objects restored	RESTORE	Sum	Null	Transfer Size	SD Name	Month
SD_TIME	SD: time in operation	SD_READ SD_WRITE	Sum	Null	Duration	SD Name	Lifetime
SD_TIME_DAY	SD: time in operation	SD_READ SD_WRITE	Sum	Null	Duration	SD Name	Day
SD_TIME_MONTH	SD: time in operation	SD_READ SD_WRITE	Sum	Null	Duration	SD Name	Month
SD_WRITE	SD: amount of data written	SD_WRITE	Sum	Null	Transfer Size	SD Name	Lifetime
SD_WRITE_DAY	SD: amount of data written	SD_WRITE	Sum	Null	Transfer Size	SD Name	Day
SD_WRITE_MONTH	SD: amount of data written	SD_WRITE	Sum	Null	Transfer Size	SD Name	Month
SD_WRITE_NUMBER	SD: number of WRITE operations	SD_WRITE	Count	Null	Event ID	SD Name	Lifetime
SD_WRITE_NUMBER_ DAY	SD: number of WRITE operations	SD_WRITE	Count	Null	Event ID	SD Name	Day
SD_WRITE_NUMBER_ MONTH	SD: number of WRITE operations	SD_WRITE	Count	Null	Event ID	SD Name	Month
TAPE_CHECKSUM_ FAILURE_COUNT_DAY	Tape: checksum failure operations count	CHECKSUM_ERROR_ TAPE TAPE_DISMOUNT_ERR TAPE_MOUNT_ERR	Count	Null	Event ID	Tape Barcode	Day
TAPE_DRIVE_DATA_ RATE	Tape Drive: data rate	TAPE_READ TAPE_WRITE	Average	Null	Transfer Rate	Drive Serial Number	Day
TAPE_DRIVE_DATA_ RATE_MONTH	Tape Drive: data rate	TAPE_READ TAPE_WRITE	Average	Null	Transfer Rate	Drive Serial Number	Month
TAPE_DRIVE_ERROR_ RATE	Tape Drive: internal error rate	TAPE_READ TAPE_WRITE	Average	Null	Error Rate	Drive Serial Number	Day
TAPE_DRIVE_ERROR_ RATE_MONTH	Tape Drive: internal error rate	TAPE_READ TAPE_WRITE	Average	Null	Error Rate	Drive Serial Number	Month
TAPE_DRIVE_LAST_ OPERATION_DATE	Tape Drive: date of last MOUNT, DISMOUNT, READ or WRITE	TAPE_DISMOUNT TAPE_MOUNT TAPE_READ TAPE_WRITE	Maximum	Null	Event Time	Drive Serial Number	Lifetime
TAPE_DRIVE_ NUMBER_MOUNTS	Tape Drive: number of mounts	TAPE_MOUNT	Count	Null	Event ID	Drive Serial Number	Lifetime

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
`TAPE_DRIVE_ NUMBER_MOUNT_ DISMOUNT_ABORTED`	Tape Drive: number of terminated MOUNT and DISMOUNT operations (together)	`TAPE_DISMOUNT_ERR` `TAPE_MOUNT_ERR`	Count	Null	Event ID	Drive Serial Number	Lifetime
`TAPE_DRIVE_ NUMBER_READ_ WRITE_ABORTED`	Tape Drive: number of terminated READ and WRITE operations (together)	`TAPE_READ_ERR` `TAPE_WRITE_ERR`	Count	Null	Event ID	Drive Serial Number	Lifetime
`TAPE_DRIVE_ NUMBER_READ_ WRITE_ABORTED_DAY`	Tape Drive: number of terminated READ and WRITE operations (together)	`TAPE_READ_ERR` `TAPE_WRITE_ERR`	Count	Null	Event ID	Drive Serial Number	Day
`TAPE_DRIVE_ NUMBER_READ_ WRITE_ABORTED_ MONTH`	Tape Drive: number of terminated READ and WRITE operations (together)	`TAPE_READ_ERR` `TAPE_WRITE_ERR`	Count	Null	Event ID	Drive Serial Number	Month
`TAPE_DRIVE_ OPERATION_TOTAL_ TIME`	Tape Drive: total time of drive operations	`TAPE_READ` `TAPE_WRITE`	Sum	Null	Duration	Drive Serial Number	Lifetime
`TAPE_DRIVE_ OPERATION_TOTAL_ TIME_DAY`	Tape Drive: total time of drive operations	`TAPE_READ` `TAPE_WRITE`	Sum	Null	Duration	Drive Serial Number	Day
`TAPE_DRIVE_READ_ WRITE`	Tape Drive: amount of data READ and written (together)	`TAPE_READ` `TAPE_WRITE`	Sum	Null	Transfer Size	Drive Serial Number	Lifetime
`TAPE_DRIVE_READ_ WRITE_DAY`	Tape Drive: amount of data READ and written (together)	`TAPE_READ` `TAPE_WRITE`	Sum	Null	Transfer Size	Drive Serial Number	Day
`TAPE_DRIVE_READ_ WRITE_MONTH`	Tape Drive: amount of data READ and written (together)	`TAPE_READ` `TAPE_WRITE`	Sum	Null	Transfer Size	Drive Serial Number	Month
`TAPE_DRIVE_READ_ WRITE_NUMBER`	Tape Drive: number of READ and WRITE operations (together)	`TAPE_READ` `TAPE_WRITE`	Count	Null	Event ID	Drive Serial Number	Lifetime

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
TAPE_DRIVE_READ_ WRITE_NUMBER_DAY	Tape Drive: number of READ and WRITE operations (together)	TAPE_READ TAPE_WRITE	Count	Null	Event ID	Drive Serial Number	Day
TAPE_DRIVE_READ_ WRITE_NUMBER_ MONTH	Tape Drive: number of READ and WRITE operations (together)	TAPE_READ TAPE_WRITE	Count	Null	Event ID	Drive Serial Number	Month
TAPE_DRIVE_TIME_ ALL_OPERATION	Tape Drive: time in all operations	TAPE_DISMOUNT TAPE_EJECT TAPE_INSERT TAPE_MOUNT TAPE_POSITION TAPE_READ TAPE_UNLOAD TAPE_WRITE	Sum	Null	Duration	Drive Serial Number	Lifetime
TAPE_DRIVE_TIME_ ALL_OPERATION_DAY	Tape Drive: time in all operations	TAPE_DISMOUNT TAPE_EJECT TAPE_INSERT TAPE_MOUNT TAPE_POSITION TAPE_READ TAPE_UNLOAD TAPE_WRITE	Sum	Null	Duration	Drive Serial Number	Day
TAPE_DRIVE_TIME_ ALL_OPERATION_ MONTH	Tape Drive: time in all operations	TAPE_DISMOUNT TAPE_EJECT TAPE_INSERT TAPE_MOUNT TAPE_POSITION TAPE_READ TAPE_UNLOAD TAPE_WRITE	Sum	Null	Duration	Drive Serial Number	Month
TAPE_DRIVE_TIME_ READ	Tape Drive: time in READ operation	TAPE_READ	Sum	Null	Duration	Drive Serial Number	Lifetime
TAPE_DRIVE_TIME_ READ_DAY	Tape Drive: time in READ operation	TAPE_READ	Sum	Null	Duration	Drive Serial Number	Day
TAPE_DRIVE_TIME_ READ_MONTH	Tape Drive: time in READ operation	TAPE_READ	Sum	Null	Duration	Drive Serial Number	Month
TAPE_DRIVE_TIME_ WRITE	Tape Drive: time in WRITE operation	TAPE_WRITE	Sum	Null	Duration	Drive Serial Number	Lifetime
TAPE_DRIVE_TIME_ WRITE_DAY	Tape Drive: time in WRITE operation	TAPE_WRITE	Sum	Null	Duration	Drive Serial Number	Day

Table B–4 (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
TAPE_DRIVE_TIME_ WRITE_MONTH	Tape Drive: time in WRITE operation	TAPE_WRITE	Sum	Null	Duration	Drive Serial Number	Month
TAPE_ EXTERNALIZATION_ NUMBER	Tape: number of externalizations	TAPE_EJECT	Count	Null	Event ID	Tape Barcode	Lifetime
TAPE_LAST_ DISMOUNT	Tape: date of last DISMOUNT	TAPE_DISMOUNT	Maximum	Null	Event Time	Tape Barcode	Lifetime
TAPE_LAST_EVENT_ ID	Tape: DIVAProtect Event ID of the last tape or drive operation	TAPE_DISMOUNT TAPE_DISMOUNT_ERR TAPE_MOUNT TAPE_MOUNT_ERR TAPE_POSITION TAPE_POSITION_ERR TAPE_READ TAPE_READ_ERR TAPE_UNLOAD TAPE_UNLOAD_ERR TAPE_WRITE TAPE_WRITE_ERR	Maximum	Null	Event ID	Tape Barcode	Lifetime
TAPE_LAST_MOUNT_ DATE	Tape: date of last MOUNT	TAPE_MOUNT	Maximum	Null	Event Time	Tape Barcode	Lifetime
TAPE_LAST_READ	Tape: date of last READ	TAPE_READ	Maximum	Null	Event Time	Tape Barcode	Lifetime
TAPE_LAST_WRITE	Tape: date of last WRITE	TAPE_WRITE	Maximum	Null	Event Time	Tape Barcode	Lifetime
TAPE_LIBRARY_ NUMBER_DISMOUNT_ ABORTED	Tape Library: total number of terminated DISMOUNT operations	TAPE_DISMOUNT_ERR	Count	Null	Event ID	Library Serial Number	Lifetime
TAPE_LIBRARY_ NUMBER_DISMOUNT_ ABORTED_DAY	Tape Library: total number of terminated DISMOUNT operations	TAPE_DISMOUNT_ERR	Count	Null	Event ID	Library Serial Number	Day
TAPE_LIBRARY_ NUMBER_DISMOUNT_ ABORTED_MONTH	Tape Library: total number of terminated DISMOUNT operations	TAPE_DISMOUNT_ERR	Count	Null	Event ID	Library Serial Number	Month
TAPE_LIBRARY_ NUMBER_MOUNT	Tape Library: total number of MOUNT operations	TAPE_MOUNT	Count	Null	Event ID	Library Serial Number	Lifetime
TAPE_LIBRARY_ NUMBER_MOUNT_ ABORTED	Tape Library: total number of terminated MOUNT operations	TAPE_MOUNT_ERR	Count	Null	Event ID	Library Serial Number	Lifetime

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
`TAPE_LIBRARY_ NUMBER_MOUNT_ ABORTED_DAY`	Tape Library: total number of terminated MOUNT operations	`TAPE_MOUNT_ERR`	Count	Null	Event ID	Library Serial Number	Day
`TAPE_LIBRARY_ NUMBER_MOUNT_ ABORTED_MONTH`	Tape Library: total number of terminated MOUNT operations	`TAPE_MOUNT_ERR`	Count	Null	Event ID	Library Serial Number	Month
`TAPE_LIBRARY_ NUMBER_MOUNT_DAY`	Tape Library: total number of MOUNT operations	`TAPE_MOUNT`	Count	Null	Event ID	Library Serial Number	Day
`TAPE_LIBRARY_ NUMBER_MOUNT_ MONTH`	Tape Library: total number of MOUNT operations	`TAPE_MOUNT`	Count	Null	Event ID	Library Serial Number	Month
`TAPE_LIBRARY_ NUMBER_READ`	Tape Library: total number of READ operations	`TAPE_READ` `TAPE_READ_ERR`	Count	Null	Event ID	Library Serial Number	Lifetime
`TAPE_LIBRARY_ NUMBER_READ_DAY`	Tape Library: total number of READ operations	`TAPE_READ` `TAPE_READ_ERR`	Count	Null	Event ID	Library Serial Number	Day
`TAPE_LIBRARY_ NUMBER_READ_MONTH`	Tape Library: total number of READ operations	`TAPE_READ` `TAPE_READ_ERR`	Count	Null	Event ID	Library Serial Number	Month
`TAPE_LIBRARY_ NUMBER_WRITE`	Tape Library: total number of WRITE operations	`TAPE_WRITE` `TAPE_WRITE_ERR`	Count	Null	Event ID	Library Serial Number	Lifetime
`TAPE_LIBRARY_ NUMBER_WRITE_DAY`	Tape Library: total number of WRITE operations	`TAPE_WRITE` `TAPE_WRITE_ERR`	Count	Null	Event ID	Library Serial Number	Day
`TAPE_LIBRARY_ NUMBER_WRITE_ MONTH`	Tape Library: total number of WRITE operations	`TAPE_WRITE` `TAPE_WRITE_ERR`	Count	Null	Event ID	Library Serial Number	Month
`TAPE_LIBRARY_READ`	Tape Library: total amount of data READ	`TAPE_READ`	Sum	Null	Transfer Size	Library Serial Number	Lifetime
`TAPE_LIBRARY_ READ_DAY`	Tape Library: total amount of data READ	`TAPE_READ`	Sum	Null	Transfer Size	Library Serial Number	Day
`TAPE_LIBRARY_ READ_MONTH`	Tape Library: total amount of data READ	`TAPE_READ`	Sum	Null	Transfer Size	Library Serial Number	Month
`TAPE_LIBRARY_ WRITE`	Tape Library: total amount of data WRITE	`TAPE_WRITE`	Sum	Null	Transfer Size	Library Serial Number	Lifetime
`TAPE_LIBRARY_ WRITE_DAY`	Tape Library: total amount of data WRITE	`TAPE_WRITE`	Sum	Null	Transfer Size	Library Serial Number	Day

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
TAPE_LIBRARY_WRITE_MONTH	Tape Library: total amount of data WRITE	TAPE_WRITE	Sum	Null	Transfer Size	Library Serial Number	Month
TAPE_MOUNT_DISMOUNT_NUMBER	Tape: number of MOUNT and DISMOUNT operations (together)	TAPE_DISMOUNT TAPE_MOUNT	Count	Null	Event ID	Tape Barcode	Lifetime
TAPE_MOUNT_NUMBER	Tape: number of MOUNT operations	TAPE_MOUNT	Count	Null	Event Id	Tape Barcode	Lifetime
TAPE_READ_WRITE_ABORTED_NUMBER	Tape: number of terminated READ and WRITE operations (together)	TAPE_READ_ERR TAPE_WRITE_ERR	Count	Null	Event ID	Tape Barcode	Lifetime
TAPE_READ_WRITE_ABORTED_NUMBER_DAY	Tape: number of terminated READ and WRITE operations (together)	TAPE_READ_ERR TAPE_WRITE_ERR	Count	Null	Event ID	Tape Barcode	Day
TAPE_READ_WRITE_NUMBER	Tape: number of READ and WRITE operations (together)	TAPE_READ TAPE_WRITE	Count	Null	Event Id	Tape Barcode	Lifetime
TAPE_READ_WRITE_NUMBER_DAY	Tape: number of READ and WRITE operations	TAPE_READ TAPE_WRITE	Count	Null	Event ID	Tape Barcode	Day
TAPE_REPACK_NUMBER	Tape: number of REPACK, REUSE, and REFORMAT operations (together)	TAPE_REPACK	Count	Null	Event ID	Local DIVArchive System	Lifetime
TRANSCODE_ABORTED_NUMBER	Transcoder: number terminated TRANSCODE operations	TRANSCODE_ERR	Count	Null	Event ID	Transcoder Name or Analyzer Name	Lifetime
TRANSCODE_ABORTED_NUMBER_DAY	Transcoder: number terminated TRANSCODE operations	TRANSCODE_ERR	Count	Null	Event ID	Transcoder Name or Analyzer Name	Day
TRANSCODE_AVG_DATA	Transcoder: average amount of data TRANSCODED	TRANSCODE_END	Weighted Average	Duration	Transfer Size	Transcoder Name or Analyzer Name	Lifetime
TRANSCODE_AVG_DATA_DAY	Transcoder: average amount of data TRANSCODED	TRANSCODE_END	Weighted Average	Duration	Transfer Size	Transcoder Name or Analyzer Name	Day

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
TRANSCODE_AVG_THROUGHPUT	Transcoder: average transcoding throughput	TRANSCODE_END	Average	Null	Transfer Rate	Transcoder Name or Analyzer Name	Lifetime
TRANSCODE_AVG_THROUGHPUT_DAY	Transcoder: average transcoding throughput	TRANSCODE_END	Average	Null	Transfer Rate	Transcoder Name or Analyzer Name	Day
TRANSCODE_DATA	Transcoder: amount of data TRANSCODED	TRANSCODE_END	Sum	Null	Transfer Size	Transcoder Name or Analyzer Name	Lifetime
TRANSCODE_DATA_DAY	Transcoder: amount of data TRANSCODED	TRANSCODE_END	Sum	Null	Transfer Size	Transcoder Name or Analyzer Name	Day
TRANSCODE_DATA_MONTH	Transcoder: amount of data TRANSCODED	TRANSCODE_END	Sum	Null	Transfer Size	Transcoder Name or Analyzer Name	Month
TRANSCODE_MAX_THROUGHPUT	Transcoder: maximum transcoding throughput	TRANSCODE_END	Maximum	Null	Transfer Rate	Transcoder Name or Analyzer Name	Lifetime
TRANSCODE_MAX_THROUGHPUT_DAY	Transcoder: maximum transcoding throughput	TRANSCODE_END	Maximum	Null	Transfer Rate	Transcoder Name or Analyzer Name	Day
TRANSCODE_MIN_THROUGHPUT	Transcoder: minimum transcoding throughput	TRANSCODE_END	Minimum	Null	Transfer Rate	Transcoder Name or Analyzer Name	Lifetime
TRANSCODE_MIN_THROUGHPUT_DAY	Transcoder: minimum transcoding throughput	TRANSCODE_END	Minimum	Null	Transfer Rate	Transcoder Name or Analyzer Name	Day
TRANSCODE_NUMBER	Transcoder: number TRANSCODE operations	TRANSCODE_END	Count	Null	Event ID	Transcoder Name or Analyzer Name	Lifetime
TRANSCODE_NUMBER_DAY	Transcoder: number TRANSCODE operations	TRANSCODE_END	Count	Null	Event Id	Transcoder Name or Analyzer Name	Day
TRANSCODE_NUMBER_MONTH	Transcoder: number TRANSCODE operations	TRANSCODE_END	Count	Null	Event ID	Transcoder Name or Analyzer Name	Month

Table B–4   (Cont.) Metrics Definitions

Metric Name	Description	Events	Operation	Weight Factor	Collection Field	Aggregation Field	Collection Interval
TRANSCODE_TIME	Transcoder: time in transcoding operations	TRANSCODE_END	Sum	Null	Duration	Transcoder Name or Analyzer Name	Lifetime
TRANSCODE_TIME_ DAY	Transcoder: time in transcoding operations	TRANSCODE_END	Sum	Null	Duration	Transcoder Name or Analyzer Name	Day
TRANSCODE_TIME_ MONTH	Transcoder: time in transcoding operations	TRANSCODE_END	Sum	Null	Duration	Transcoder Name or Analyzer Name	Month

Configuration Parameter Defaults and Values

Table B–5   Configuration Parameter Defaults and Values

Parameter	Default	Values
Manager: Enable/Disable DIVAprotect Data Collection	1	0 or 1
Manager: Size of the event batch download (number of events)	100	Integer
Manager: Max timeout in the event there are not events to fill the above batch (seconds)	15	Integer
Conf Utility GUI: Enable/Disable DIVAprotect Configuration	0	0 or 1
DB: Maximum possible number of Events in DB	1000000	Integer
DB: Maximum possible number of Metrics in DB	1000000	Integer

C

Sources and Destinations Guide

This appendix describes Source and Destination configuration guidelines for each type of DIVArchive supported content server. See the *Oracle DIVArchive Supported Environments Guide* documentation in the *Oracle DIVArchive Core documentation* library for detailed and up-to-date lists of supported content servers, formats, and related DIVArchive platforms.

The following information is included:

- General Parameters
  - Files Path Root Parameter
  - Root Path Parameter
    - * UNIX Style Paths
    - * Windows Style Paths
  - Metasource Parameter
  - Connect Options Parameter
    - * Quality of Service (`qos=`)
    - * Source/Destination FTP User Log In (`-login`)
    - * Source/Destination CIFS User Log In (`-user`)
    - * Source/Destination Password (`-pass`)
    - * Source/Destination Connection Port (`-port`)
    - * Deleting Source Content after Archiving (`-allow_delete_on_source`)
    - * Archiving and Restoring File Renaming Rules (`-arch_renaming`, `-rest_renaming`)
    - * Skipping Files During Restore (`-rest_ignoring`)
    - * Archiving Files in a Specific Order (`-file_order`)
    - * Specifying the Transcode Format (`-tr_archive_format`, `-tr_restore_format`)
    - * Specifying a Transcoder Name (`-tr_names`)
    - * Restoring Metadata (`-rest_metadata`, `-rm`)
    - * Restricting the Number of Actors to Retry (`-num_actors_to_retry`)
    - * MSS Source/Destination in MXF Mode (`-mxf`)
    - * FTP Socket Window Size (`-socket_window_size`)

General Parameters

this section introduces general items that may apply to any, or most, Sources or Destinations including features, configuration attributes, and connection options.

Files Path Root Parameter

The *Files Path Root* (FPR) parameter is for Archive and Restore requests. This parameter specifies the root folder for data transfers and applies to any type of Source/Destination.

You can enter an absolute or relative path in the *Files Path Root* field. This parameter is limited to 260 characters.

Each content server section of this appendix specifies the expected format of the *Files Path Root* and related *File Names* parameters for Archive requests.

For Partial File Restore requests, the file names on the destination are those specified when archiving. If no *Files Path Root* is entered, DIVArchive uses the one specified during archiving.

Root Path Parameter

The *Root Path* is a Source/Destination attribute you can use as a default path for FTP-like Sources/Destinations, or as a disk mount point for disk and local sources. This applies to any type of Source/Destination. The path is appended before any *Files Path Root* specified in requests, unless the path specified in a request is an absolute path.

This approach provides better Source/Destination abstraction. You specify the server directories used by DIVArchive at the configuration level, not at the request level. They can be changed at any time without requiring a change to DIVArchive clients.

The *Root Path* value is always an absolute path defined by the operating system. An Omneon Path is the player name and always considered an absolute path.

Absolute path names are supported on both Windows and Linux to a maximum of 4000 characters. Relative path names are limited to 256 characters on Windows systems (only).

If you leave the *Root Path* field empty, DIVArchive ignores the parameter. However, if you do specify a *Root Path* its value is combined with the *Files Path Root* you specified in a request to give the final Source/Destination path. This process is performed according to the following rules:

- Relative paths are added to the absolute path, absolute paths override preceding absolute paths (standard *Path Arithmetic*).

- If the *Root Path* and *Files Path Root* have different operating system types, the second path (*Files Path Root*) is converted to the operating system type specified by the first path (*Root Path*) by replacing \ with / (and vice versa). The converted path is then considered the relative path.

- If the *Root Path* ends with a > character, the *Files Path Root* is always considered to be a relative path, and the > character is omitted during concatenation.

Table C–1   Root Path Definitions

Source/Destination `ROOT_PATH`	Object: `Original_FPR` recorded in database & metadata	Request Type	Files Path Root (FPR)	Resulting rule applied to create actual path for the transfer	Resulting path considered for the transfer	Resulting original Files Path Root (FPR) recorded in database and metadata
Null		Archive	Null	`ROOT_PATH+FPR`	Null	Null
Null		Archive	Set	`ROOT_PATH+FPR`	FPR	FPR
Set		Archive	Null	`ROOT_PATH+FPR`	`ROOT_PATH`	Null
Set		Archive	Set	`ROOT_PATH+FPR`	`ROOT_PATH+FPR`	FPR
Null		Archive with `tr_arch` format	Null	`ROOT_PATH+FPR`	Null	Null
Null		Archive with `tr_arch` format	Set	`ROOT_PATH+FPR`	FPR	Null

Table C–1   (Cont.) Root Path Definitions

Source/Destination ROOT_PATH	Object: Original_FPR recorded in database & metadata	Request Type	Files Path Root (FPR)	Resulting rule applied to create actual path for the transfer	Resulting path considered for the transfer	Resulting original Files Path Root (FPR) recorded in database and metadata
Set		Archive with tr_arch format	Null	ROOT_PATH+FPR	ROOT_PATH	Null
Set		Archive with tr_arch format	Set	ROOT_PATH+FPR	ROOT_PATH+FPR	Null
Null	Null	Restore	Null	(ROOT_PATH+FPR) | Original_FPR	Null	
Null	Null	Restore	Set	(ROOT_PATH+FPR) | Original_FPR	FPR	
Set	Null	Restore	Null	(ROOT_PATH+FPR) | Original_FPR	ROOT_PATH	
Set	Null		Set		ROOT_PATH+FPR	
Null	Set		Null		Original FPR	
Null	Set		Set		FPR	
Set	Set		Null		ROOT_PATH	
Set	Set		Set		ROOT_PATH+FPR	
	Null	Transcode Archive				Null
	Set	Transcode Archive				Null

UNIX Style Paths

The following table describes UNIX style paths for the *Root Path*, *File Path Root*, and the actual path to the files.

Table C–2   UNIX Style Paths

Root Path (Source/Destination)	File Path Root (Request)	Actual Path to Files
/diva/upload	tmp	/diva/upload/tmp
/diva/upload	/tmp	/tmp
/diva/upload		/diva/upload
/diva/upload	C:\tmp	/diva/upload/C:/tmp (*!!!*)
/diva/upload>	/tmp	/diva/upload/tmp
/diva/upload>	\tmp	/diva/upload/tmp
/diva/upload>		/diva/upload

Windows Style Paths

The following table describes Windows style paths for the *Root Path*, *File Path Root*, and the actual path to the files.

Table C–3   Windows Style Paths

Root Path (Source/Destination)	File Path Root (Request)	Actual Path to Files
D:\diva\upload	tmp	D:\diva\upload\tmp
D:\diva\upload	C:\tmp	C:\tmp
D:\diva\upload		D:\diva\upload
D:\diva\upload>	/tmp	D:\diva\upload\tmp
D:\diva\upload>	C:\tmp	D:\diva\upload\tmp
D:\diva\upload>	C:/tmp	D:\diva\upload\C:\tmp
D:\diva\upload>		D:\diva\upload

Metasource Parameter

The *Metasource* parameter is a specific type of Source/Destination to manage several Sources/Destinations sharing the same online storage as one (or multiple Drop Folder Monitors) with failover and load-balancing features. This applies to any type of Source/Destination. See Metasources for more information on the Metasource Source/Destination types.

Connect Options Parameter

Connect Options are a Source/Destination parameter used to specify the communication protocol with the Source/Destination or to modify the protocol's defaults.

Some options exclusively apply to a specific Source/Destination type, and are documented as part of that specific Source/Destination type later in this appendix. Others options are for general use and are documented in this section.

Some *Connect Options* (explicitly or implicitly) specified for the Source/Destination may be superseded by those specified in requests. This section also specifies, for each *Connect Option*, whether it can be superseded at the request level.

Quality of Service (qos=)

This option specifies the transfer mode used when transferring from this specific Source/Destination when the archive initiator sets the QualityOfService parameter in Archive or Restore parameters to **DEFAULT**.

This parameter applies to any type of Source/Destination, and cannot be superseded by the request option.

If the archive initiator sets the QualityOfService to something other than **DEFAULT**, DIVArchive ignores the qos= *Connect Option*.

The format for the parameter is qos=[**DIRECT_AND_CACHE**|**CACHE_AND_DIRECT**].

> **Note:** This option must be the first one in place in the *Source/Destination Connect Options* field, and must *always* be specified in lowercase.

The valid values for Quality of Service are as follows:

DIRECT_AND_CACHE
Direct transfers from (or to) a Source/Destination to (or from) DIVArchive are preferred, but cache transfers will occur if processing the request in direct mode is not possible.

CACHE_AND_DIRECT
Cache transfers from (or to) a Source/Destination to (or from) DIVArchive are preferred, but direct transfers will occur if processing the request in cache mode is not possible.

The following table describes sample Quality of Service connections:

Table C–4   Sample Quality of Service Connections

QOS Connect Option	QOS Set by the Archive Initiator	Actual Transfer Mode Applied by the DIVArchive Manager
DIRECT_AND_CACHE	DEFAULT	DIRECT_AND_CACHE
DIRECT_AND_CACHE	DIRECT_ONLY	DIRECT_ONLY
DIRECT_AND_CACHE	CACHE_ONLY	CACHE_ONLY
CACHE_AND_DIRECT	DEFAULT	CACHE_AND_DIRECT
CACHE_AND_DIRECT	DIRECT_ONLY	DIRECT_ONLY
CACHE_AND_DIRECT	CACHE_ONLY	CACHE_ONLY
	DEFAULT	**DEFAULT** (that is, **DIRECT_AND_CACHE**)
	DIRECT_ONLY	DIRECT_ONLY
	CACHE_ONLY	CACHE_ONLY

Source/Destination FTP User Log In (`-login`)

This option is generally used to specify a user name to connect to a Source/Destination when the transfer protocol is FTP or FTP-like, and is in the format `-login {user_name}`.

This option applies when specified in Source/Destination type description, and can be superseded by the request option.

Possible values applicable to a specific Source/Destination type are detailed in the related section later in this appendix.

Source/Destination CIFS User Log In (`-user`)

This option is generally used to specify a user name to connect to a CIFS Source/Destination, and is in the format `-user {user_name@domain}`.

This option applies when specified in Source/Destination type description, and can be superseded by the request option.

Possible values applicable to a specific Source/Destination type are detailed in the related section later in this appendix.

Source/Destination Password (`-pass`)

This option is generally used in combination with the `-login` option, and is in the format `-pass [password]`.

This option applies when specified in Source/Destination type description, and can be superseded by the request option.

Possible values applicable to a specific Source/Destination type are detailed in the related section later in this appendix.

Source/Destination Connection Port (`-port`)

This option is used when a port parameter is required to connect to a Source/Destination, and specifies the port number in the format `-port [port_number]`.

This is an integer value that applies when specified in Source/Destination type description, and can be superseded by the request option.

Possible values applicable to a specific Source/Destination type are detailed in the related section later in this appendix.

Deleting Source Content after Archiving (`-allow_delete_on_source`)

This parameter specifies if an Archive request can use the *Delete on Source* QOS option, and is in the format `-allow_delete_on_source`.

The Archive request optional parameter `delete_on_source` instructs DIVArchive to delete the original asset on the source after the archive of the object is successfully completed.

If this option is specified in an Archive request and the *Source Type* is not **LOCAL**, **DISK** or **CIFS**, DIVArchive automatically terminates the request.

This parameter applies to the **FTP_STANDARD** *Source Type*. you can change this behavior so that requests will not fail when `delete_on_source` is specified in an Archive request.

If the `-allow_delete_on_source` option is specified, and the `delete_on_source` parameter is specified in an Archive request, DIVArchive will attempt to delete the asset from the source after the archive has been completed successfully.

This option cannot be superseded by the request option.

Archiving and Restoring File Renaming Rules (`-arch_renaming`, `-rest_renaming`)

This feature is available for Archive and Restore requests. There are no pre-defined set of values for these options. Option values are based on regular expressions. Possible values for these options are infinite and fully customizable.

Renaming rules are associated with Source/Destination. You configure file renaming during archive or restore using the Configuration Utility. The configuration can be superseded by the request option.

You use these parameters when a workflow implementation requires automatic file renaming during object archiving, when the object is (partially) restored, or when a transcoded object is rearchived or restored.

Rename files at archive time (`-arch_renaming`) or at restore time (`-rest_renaming`). The format for these parameters are as follows:

```
-arch_renaming [renaming_rule]+
-rest_renaming [renaming_rule]+

renaming_rule = [activation_format:expression_patterns:output_format]
```

The `-arch_renaming` option enables renaming files during the archive process. You typically use this option for the following example cases:

- You must add a file extension to archived files.

- When associated to a transcoder cache (Local Source/Destination), you can set archive renaming rules to rename the files of a transcoded clip. This is useful when files created by the transcoder do not have the expected names.

The `-rest_renaming` option enables renaming of files during the restore process. You typically use it when the Source/Destination requires strict naming of files, and the files being transferred do not follow these rules.

This option is available for **Restore**, **Partial File Restore** (this is an alternate way to rename partially restored files), and **N-Restore**. If multiple renaming rules are defined, DIVArchive will process the rule for each Source/Destination independently.

You must specify at least one `renaming_rule` for the option. All renaming rules are located in the Configuration Utility except the *Service Name* and *Port* parameters. DIVArchive goes through each `renaming_rule` for each file on the list to be transferred starting with the first one:

- The rule is applied if the file name matches this rule's `activation_format`.

- The condition is satisfied if the beginning of a file name matches the evaluation condition of the first rule.

  For example, a condition such as `.*\.track` will be satisfied by all of the following file names - `audio.track1`, `audio.track2`, `video.track`.

- As soon as a rule is applied for a given file, other rules from the list are no longer considered.

- If none of the rules can be applied, the file is not renamed.

An `activation_format` is a regular expression (regexp).

The `expression_patterns` parse the file name. It is a regular expression, which will include up to nine special symbols to identify different parts of the file name: `\1 \2 \3 \4 \5 \6 \7 \8 \9`.

The `output_format` is an expression that qualifies the format of a renamed file, based on atomic items (`\1` through `\9`) previously identified when applying `expression_patterns` to the original file name. Two additional specific symbols can be used:

- `\o` indicates the object name

- `\c` indicates the object category

> **Note:** Describing how regular expression pattern matching works is beyond the scope of this document. There are many web sites on this subject such as http://www.regular-expressions.info/.

The following examples describe different possible scenarios and their associated outcomes using these parameters.

Example One

To add the `.gxf` extension to all files archived from GVG Profile (by default, these files do not have an extension). If a file does have an extension, the `.gxf` extension will not be added. To achieve this you use the following statement:

```
-arch_renaming <.*\..*:(.*)\.(.*):\1.\2><.*:(.*):\1.gxf>
```

This statement will process the file names as follows:

Input file Name	Output File Name
Star Wars	Star Wars.gxf
Readme.txt	Readme.txt
Jaws.gxf	Jaws.gxf

Example Two

To remove the `.gxf` extension (if any) at archive time you use the following statement:

`-arch_renaming <.*\.gxf:(.*)\.(.*):\1>`

This statement will process the file names as follows:

Input File Name	Output File Name
Star Wars.gxf	Star Wars
Readme.txt	Readme.txt
Jaws.avi	Jaws.avi

Example Three

When Flip Factory transcodes clip `FOO` to Pinnacle MSS, the resulting files are named `FOO.MSS.header`, `FOO.MSS.ft`, `FOO.MSS.info`, and `FOO.MSS`. These names are not those expected by Pinnacle MSS Servers, and this option fixes these discrepancies. You use the following statement:

`-arch_renaming`
`<.*\.header:(.*):header><.*\.ft:(.*):ft><.*\.info:(.*):info><.*MSS:(.*):std>`

This option will process the file names as follows:

Input File Name	Output File Name
FOO.MSS.header	header
FOO.MSS.ft	ft
FOO.MSS.info	info
FOO.MSS	std

To help regular expression development, regular expression exercisers are available online at http://regexone.com/ and http://www.lornajane.net/posts/2011/simple-regular-expressions-by-example.

To use this feature, you must know the basic regular expression syntax. You can find Regular Expression introductory information online at http://www.hathitrust.org/, http://books.google.com/, and http://www.gutenberg.org/.

Skipping Files During Restore (`-rest_ignoring`)

This option is available for Restore, Partial File Restore, and N-Restore requests. It ignores some files during restore based on one or more regular expression rules. The possibilities offered by regular expressions are versatile and enable many different types of filtering.

Files matching one of the regular expressions are ignored by the Source/Destination. The rule supports Unicode characters to offer maximum flexibility. You use the following statement to ignore files during restore:

```
-rest_ignoring {<rule>} [<rule>|<rule>|<rule>]
```

You can continue to add `<rules>` as necessary in the previous statement.

There are no predefined set of values for these options. Possible values for this option are infinite and fully customizable.

The files being ignored are still read from the disk or tape instance. If the set of rules is designed to ignore all the files of an object, then no file is restored and the request will be complete.

During an **N-Restore**, if multiple renaming rules are defined, DIVArchive will process the rule for each Source/Destination independently.

Example

A typical use case is restoring a SeaChange clip to a destination that does not support SeaChange special files (private data and video index files). The following statement prevents a Source/Destination from restoring files with `.pd` or `.vix` extension:

```
-rest_ignoring <.*\.pd><.*\.vix>
```

The results if the previous statement are as follows:

DIVArchive Object	Destination Server
Clipname.pd	
Clipname.vix	
Clipname	Clipname

Archiving Files in a Specific Order (`-file_order`)

You use this option archiving or restoring files that are MSS files (Omneon QuickTime files), but the source of archiving is not an AVID (Pinnacle) MSS Server (an Omneon server).

This option is not limited to specific Source/Destination types, but it is only meaningful for **LOCAL**, **DISK**, **CIFS**, and **FTP_STANDARD** Source/Destinations. This option can be superseded by the request option.

You specify the file sequence during archiving or restoring using the following statement:

```
-file_order {MSS|OMNEON|DIFWAV|SEACHANGE DIRS_FIRST|FILES_FIRST}
```

The following list identifies the archive sequence for specific formats:

MSS
The sequence is `header`, `ft`, `info`, and then `std`.

OMNEON
The sequence is `clip.mov`, and then essence files (`.wav`, `.aiff`, `.m2v`, `.mpeg`, `.diff`, and so on).

DIFWAV
The sequence is `clip.dif`, and then `.wav` files.

SEACHANGE

The sequence is `clip.pd`, `clip.vix`, and then `clip`.

DIRS_FIRST

The sequence places directories first and is as follows:

```
Folder test_1;
Folder test_1\test_2;
File test_1\test_2\1.txt;
File test_1\test_2\_A2.txt;
File test_1\test_2\test.txt;
File test_1\test_2\test1.txt;
File test_1\test_2\test2.txt;
File test_1\1.txt;
File test_1\_A2.txt;
File test_1\test.txt;
File test_1\test1.txt;
File test_1\test2.txt;
File 1.txt;
File _A2.txt;
File test.txt;
File test1.txt;
File test2.txt;
```

FILES_FIRST

The sequence places files first and is as follows:

```
File 1.txt;
File _A2.txt;
File test.txt;
File test1.txt;
File test2.txt;
Folder test_1;
File test_1\1.txt;
File test_1\_A2.txt;
File test_1\test.txt;
File test_1\test1.txt;
File test_1\test2.txt;
Folder test_1\test_2;
File test_1\test_2\1.txt;
File test_1\test_2\_A2.txt;
File test_1\test_2\test.txt;
File test_1\test_2\test1.txt;
File test_1\test_2\test2.txt;
```

This ensures that files are archived in the correct sequence so that they are restored in the correct sequence when restoring them to a real Pinnacle MSS Server (a real Omneon server).

DPX Partial File Restore does not examine the file name or the DPX header information to determine which file is assigned to which frame. The assignment is based purely on the sequence in which the `.dpx` files appear within the archive. By default this sequence is based on ordering established by the source, and is typically alphanumeric. For example, NTFS **DISK** Source/Destinations order files and folders are not case-sensitive as a general rule (but not where diacritical marks, such as ', `, ^, and so on are applied). By default, when DIVArchive encounters a subfolder it recursively processes all of the children of that folder (including subfolders) before continuing with other files. If a folder appears in the alphanumeric folder listing, it is archived recursively in the order it appears.

However, this can create some issues. You may want all of the subdirectories of a given directory processed first, followed by the files in the directory. Or, you might want all files processed first, then subdirectories. In DIVArchive 7.0 and later, the Actor allows the archive options -file_order DIRS_FIRST or -file_order FILES_FIRST to address these issues as previously described.

Example

An archive contains SeaChange SAF files. These files must be transcoded, and then restored to a Pinnacle MSS Server. In this case, the **LOCAL** source used by the transcoding process is defined with the -file_order MSS option (among others).

This ensures the files coming out of the transcoder are archived and restored in the correct sequence. That is, header, ft, info and then std.

Specifying the Transcode Format (`-tr_archive_format`, `-tr_restore_format`)

Each factory in a transcoder determines the format of the output file. These options allow you to define the factory and output format.

They apply to any Source/Destination type, and have no fixed list of values. This option cannot be superseded by the request option unless used in a TranscodeArchived request.

These options specify the transcode operation to apply to essence files during archive (-tr_archive_format) or restore (-tr_restore_format).

```
-tr_archive_format {factory_name}
-tr_restore_format {factory_name}
```

The {factory_name} is the name of a Flip Factory factory, or the name of a Bitscream output format.

Specifying a Transcoder Name (`-tr_names`)

You use this option to specify the transcoder to use for transcode operations. It applies to any Source/Destination type and cannot be superseded by the request option, unless used in a TranscodeArchived request.

You must always use either the -tr_archive_format or the -tr_restore_format option with -tr_names. When transcoding is applied, one of the transcoders defined by -tr_names is selected by DIVArchive according to the transcoders defined in the DIVArchive configuration based on the availability, configured queue size, and configured performance of the transcoder.

The format for this option is as follows:

```
-tr_names {transcoder_name} [transcoder_name]
```

The {transcoder_name} is the name of a DIVArchive Transcoder defined in the *Transcoders* frame of the **Systems** tab of the Configuration Utility.

If this option is not present, DIVArchive will select one of the transcoders defined in the DIVArchive Configuration based on the availability, configured queue size, and configured performance of the transcoder.

Restoring Metadata (`-rest_metadata`, `-rm`)

This option specifies that a metadata file must be generated and restored on every Restore request. This option applies to any Source/Destination type. Because video servers may reject the metadata file, this option actually applies mainly to **LOCAL**, **DISK** and **FTP_STANDARD** types.

Either form of the option can be used as follows:

```
-rest_metadata
-rm
```

When and object is restored, the object is first restored normally. After the regular restore has completed, a metadata file is generated and restored on the specified destination in the specified (or implicit) *FilePathRoot* of the related Restore request.

The metadata file format is compliant with the *DIVArchive File Set Drop Folder Metadata File* specification. The metadata file name is `objectname.mdf`.

Restricting the Number of Actors to Retry (`-num_actors_to_retry`)

You use this option to limit the number of Actors that an Archive, Restore, or Partial File Restore request will be retried on. By default, this option is not specified and there is no limit. Therefore, all Actors will be tried in case the request constantly fails.

This option applies to any Source/Destination type and cannot be superseded by the request option.

This option uses the following statement:

```
-num_actors_to_retry {number}
```

The {number} is the number of retries to attempt and can include zero.

Example

The option `-num_actors_to_retry` 3 means that the DIVArchive Manager will perform no more than four operations (total) with different Actors, even if there are more than four Actors configured. That is, the initial request plus three retries for a total of four attempts.

MSS Source/Destination in MXF Mode (`-mxf`)

This option specifically applies only to MSS Source/Destination types, otherwise DIVArchive ignores it. You use this option to indicate when a MSS Source/Destination is configured to import and export MXF wrapped clips.

There are no additional parameters for this option and you include it in the following format:

```
-mxf
```

FTP Socket Window Size (`-socket_window_size`)

This option specifies the total buffer space per data socket reserved for send and receive. This option applies to some Source/Destination types using FTP protocol, such as **FTP_STANDARD**, **OMNEON**, **PDR**, **MSS**, and so on.

This parameter has a direct effect on transfer performance. Its value depends on the operating system and is usually set between 2048 and 65536 bytes. When this option is not set DIVArchive uses the default value set at the operating system level. Oracle recommends increasing this value to 32768 or more on fast networks. You must run some performance tests to identify the best setting.

*The **TCP Window Scale** option increases the TCP receive window size above its maximum 65536 bytes value. This option is recommended when dealing with Long-Fat Networks, or LFN.*

You use the following statement for this option:

```
-socket_window_size {number}
-socket_bufsize {number}
```

The `{number}` is the buffer size in bytes.

> **Note:** The `-socket_bufsize` syntax deprecated but still available.
> Oracle recommends not using it in DIVArchive releases later than 6.2.2
> because it may conflict with the `-socket_block_size` parameter.

FTP Socket Block Size (`-socket_block_size`)

This option defines how much data (in kilobytes) the Actor tries to send and receive in a single system call during FTP transfers. For example, if the internal buffer size of the Actor is set to 2 Mb and `-socket_block_size` is set to 64 KB, 32 system calls are required to write a single buffer to a data socket.

This option applies to some Source/Destination types using FTP, such as **FTP_STANDARD**, **OMNEON**, **PDR**, **MSS**, and so on.

You use the following statement for this option:

```
-socket_block_size {number}
```

The `{number}` is the buffer size in kilobytes, ranging from 32 to 2048 kilobytes.

FTP Passive Mode Transfers (`-pasv`)

This option specifies that the FTP data connection must be opened in passive mode (as opposed to active mode) for the associated Source/Destination. This may be necessary if a firewall is between the Actor and the Source/Destination.

This option applies to some Source/Destination types using FTP, such as **FTP_STANDARD**, **OMNEON**, **PDR**, **MSS**, and so on.

You use one of the following statements for this option (not case-sensitive):

```
-pasv
-PASV
```

Restoring in AXF Mode (`-axf`)

The `-axf` parameter is optional for Restore requests and instructs DIVArchive to restore the original asset into an AXF File. Instead of purely restoring the content of an object to the destination, DIVArchive restores the content into a new AXF File.

Combined with the `-rm` or `-rmxl` parameters, you can use this option to export an object with metadata information and then drop it into a DFM Watch Folder.

This option applies to **FTP_STANDARD**, **SFTP**, **LOCAL**, **DISK**, and **EXPEDAT** Source/Destination types.

You use the following statement to restore an asset in AXF mode:

```
-axf
```

Specifying Connection Timeouts (`-list_timeout`, `-transfer_timeout`, `-control_timeout`)

These options specify the maximum timeout values allowed for different FTP connection operations, and override the default timeout settings. You can set the

timeout value for directory and file listings (-list_timeout), file transfers (-transfer_timeout), and control port connections (-control_timeout).

If an operation exceeds the set timeout value the operation is terminated.

The default value is used if a timeout parameter is not specified, or if the timeout value is set to zero.

You use the following statement for each of these options:

```
-list_timeout {number}
-transfer_timeout {number}
-control_timeout {number}
```

The {number} is the maximum allowed timeout in seconds.

The default timeout values for each FTP connect operation are as follows:

Statement	Default Timeout
-list_timeout	120 seconds
-transfer_timeout	180 seconds
-control_timeout	120 second

Avid MSS (Program Stream) Servers

Avid (previously Pinnacle) MSS Video Servers can be installed in one of the following configurations:

Independent Storage
The video server (itself) includes its own fault tolerant disk storage.

Shared Storage
The video servers are connected to a SAN where the fault tolerant disk storage is based.

In both cases, external connectivity is provided by one (or several) Connect+ gateways supporting the FTP protocol over a Gigabit Ethernet Network. A clip on the MSS storage is always comprised of three files as listed below (or four if the optional information file is present). They are always archived and restored in the following sequence:

header
This is the first file and the clip's header.

ft
This is the second file and the frame table.

std
This is the third file and the video and audio essence.

info
When present, this is the fourth file. It is an optional information file.

All files are located in a folder that matches the name of the clip (that is, if the clip name is FOO, the files are located in a folder also named FOO).

Newer MediaStream servers can export and import clips with a MXF wrapper. When configured for MXF, the server generates a single file (std) which is the MXF file. DIVArchive only archives one file (std) in MXF Mode. The file is automatically

renamed to `{clipname}.mxf`. This mode is not supported by independent storage servers.

See Appendix A for Oracle DIVArchive options and licensing information.

MSS with Independent Storage

One record is created for each MSS that DIVArchive must move data to and from.

Table C–5   MSS with Independent Storage Parameters

Attribute	Value	Example
IP Address	MSS IP address	`10.80.114.21`
Source Type	MSS	**MSS**

Connect Options for Systems with One Gateway	`-login {gw_host_name}`
	
	`-pass .video_fs`
		`-login fcgate1`
		
		`-pass .video_fs`

Connect Options for Systems with Two Gateways	`-login {gw1_host_name}[, gw2_host_`
	`name]`
	
	`-pass .video_fs`
		`-login fcgate1,fcgate2`
		
		`-pass .video_fs`

In a system with two gateways, `fcgate1` and `fcgate2`, DIVArchive manages failover between the two when a connect option such as `-login fcgate1, fcgate2` is specified. If the initial FTP connection fails with `fcgate1`, it will be retried on `fcgate2`.

*This feature has been deprecated and is now implemented using the **METASOURCE** Source Type.*

MSS with Shared Storage

One record is created for each gateway connected to the storage network that DIVArchive must move data to and from.

Table C–6   MSS with Shared Storage Parameters

Attribute	Value	Example
IP Address	IP Address of the gateway through which DIVArchive accesses the shared storage.	`10.80.114.28`
Source Type	MSS	**MSS**

Connect Options	`-login video_fs`
	
	`-pass .video_fs`
		`-login video_fs`
		
		`-pass .video_fs`

MSS with Shared Storage in MXF Mode

One record is created for each gateway connected to the storage network DIVArchive has to move data to and from.

Table C–7   MSS with Shared Storage in MXF Mode Parameters

Attribute	Value	Example
IP Address	IP Address of the gateway through which DIVArchive accesses the shared storage.	`10.80.114.28`
Source Type	MSS	**MSS**

Table C–7   (Cont.) MSS with Shared Storage in MXF Mode Parameters

Attribute	Value	Example
Connect Options	`-login video_fs` (or `-login mxf_fs`) `-pass .video_fs` (or `-pass .mxf_fs`) `-mxf`	`-login video_fs` `-pass .video_fs`

Using MSS with `DIVA_archiveObject`

The following table describes typical Source/Destination example parameters.

Table C–8   `DIVA_archiveObject` Source/Destination Use Example

Parameter	Value	Example
FilesPathRoot	The name of the clip.	`CITIZENKANE`
FileNames	*	*

Avid Airspace Servers

Avid Airspace (previously known as Pluto) is a video server with independent storage. Each clip deals with a single essence file located on the storage root. Airspace uses standard FTP protocol to transfer files to and from the video server internal storage over a Gigabit Ethernet Network.

One record is created for each video server DIVArchive has to move data to and from.

Table C–9   Avid Airspace Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the video server.	`10.80.114.28`
Source Type	**FTP_STANDARD**	`FTP_STANDARD`

Connect Options	`-login {FTP_user_name}`
	 `-pass {FTP_password}`
	 `-port {FTP_port_number}`
		`-login ftpuser`
		 `-pass Pa$$word`
		 `-port 6530`

The following table describes an Avid Airspace Source/Destination use example:

Table C–10   Avid Airspace `DIVA_archiveObject` Source/Destination Use Example

`DIVA_archiveObject` Parameter	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	Enter the name of the clip in this field.	`TRAFFIC`

Avid Transfer Manager DHM Interface

The Avid Transfer Manager is the Avid Unity Outer Gateway, which you can address using two different interfaces. One interface is called the Data Handler Module (DHM) and the other called Dynamically Extensible Transfer (DET). Each interface has a specific purpose.

For this *Source Type* the DHM interface is used for transfer of video and audio content to and from external devices (for example, an archive system).

See the *Oracle DIVArchive Avid Connectivity User's Guide* in the *Oracle DIVArchive Additional Features documentation* library for detailed information.

One record is created for each video server DIVArchive has to move data to and from.

Table C–11   Avid DHM Source/Destination Attributes

Attributes	Value	Example
IP Address	PI address of the Avid Transfer Manager	`10.80.114.28`
Source Type	**AVID_DHM**	`AVID_DHM`

Connect Options	`-port {FTP_port_number}`
	`-login {FTP_user_name}`
	`-pass {FTP_password}`
		`-port 6021`
		`-login diva`
		`-pass diva`

The *Connect Option* values indicated in the previous table are as follows:

-port
This is the TM Communicator FTP service port number.

-login
This is the TM Communicator FTP service user log in.

-pass
This is the TM Communicator FTP service user password associated with the log in.

Archive requests are initiated from Avid Edit Stations using **Send to Playback**. The TM Communicator supports setting custom titles for ingested (restored) clips. If the `-title` option is specified with a title name in a DIVArchive Restore or Partial File Restore request, this option's value is used as the clip title, otherwise the original clip name is used. The original clip name is stored in the *Video ID* field of the Avid metadata.

The following rules apply to custom title settings:

- Custom titles can consist of one or more words separated by spaces and (or) tabulation characters.
- Oracle strongly recommends single word titles, and absolutely requires that multiple word titles are enclosed in double quotes to ensure proper processing.
- New line (`\x0A`) and carriage return (`\x0D`) characters are not allowed in titles.
- Single quote, ampersand, dash, slash, asterisk, and other special characters are supported.
- Double quote characters must be escaped with a backslash to be included in the title.
- Titles composed of one or more spaces enclosed in double quotes are not considered empty.

The following table describes a Source/Destination use example:

Table C–12   Avid DHM Source/Destination Use Example

Restore Option Values	Ingested Clip Title
`-title Clip`	`Clip`
`-title "Clip"`	`Clip`

Table C–12   (Cont.) Avid DHM Source/Destination Use Example

Restore Option Values	Ingested Clip Title
`-title "My clip"`	`My clip`
`-title "My \"special\" clip"`	`My "special" clip`

Avid Transfer Manager DET Interface

Avid Transfer Manager is the Avid Unity Outer Gateway. It can be addressed through two different interfaces called the Data Handler Module (DHM) and Dynamically Extensible Transfer (DET). Each interface has a specific purpose.

For this source type, the DET interface is used for transfer of Metadata and Media Files to Unity Workgroups (or an archive system, seen as an external workgroup / Unity storage extender).

See the *Oracle DIVArchive Avid Connectivity User's Guide* in the *Oracle DIVArchive Additional Features documentation* library for detailed information.

One record is created for each video server DIVArchive has to move data to and from.

Table C–13   Avid DET Source/Destination Attributes

Attributes	Value	Example
IP Address	IP address of the Avid Transfer Manager	`10.80.114.28`
Source Type	**AVID_DET**	**`AVID_DET`**

Connect Options	`-port {FTP_port_number}`
	`-login {FTP_user_name}`
	`-pass {FTP_password}`
		`-port 6021`
		`-login det`
		`-pass diva`

The *Connect Option* values indicated in the previous table are as follows:

`-port`
This is the TM Communicator FTP service port number.

`-login`
This is the TM Communicator FTP service user log in.

`-pass`
This is the TM Communicator FTP service user password associated with the log in.

Archive requests are initiated from Avid Edit Stations using **Send to Workgroup**.

SeaChange BMS and BMC Servers

A SeaChange BMS (Broadcast Media Server) is a standalone video server equipped with a fast-Ethernet Interface and its own storage.

A SeaChange BMC (Broadcast Media Cluster) is a cluster of video servers providing unified storage based on SeaChange RAID^2 technology. Each server of the BMC can deliver files stored on RAID^2 to DIVArchive using the FTP protocol. The file transfer format is SAF (SeaChange Archive Format) only.

> **Note:**   The SeaChange FTP servers do not support directories. All files must be listed under the FTP root directory.

By default, a SeaChange BMC node offers Automatic Load Balancing management for data transfer across all nodes of the cluster.

If you want to use this feature, you must only declare the last node of the BMC in the DIVArchive configuration. In this case, DIVArchive will always connect to the same node of the cluster. This node will transparently redirect transfers to other nodes as required.

This feature can be disabled by using a special IP address setting in the DIVArchive configuration (see the following table). In this case, *all* nodes of the BMC must be declared in the DIVArchive configuration.

You can also add a Metasource that encompasses all nodes of the cluster to enable load balancing and failover from within DIVArchive.

Table C–14   SeaChange BMS and BMC Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the BMS or BMC node.	`10.80.114.26`
	You can disable the SeaChange Automatic Load Balancing by placing a $ in front of the IP address of *all* BMC nodes. The syntax for this is `$IP_Address`.	`$10.80.114.26`
Source Type	**SEACHANGE_BMC**	`SEACHANGE_BMC`
DIVAACTOR_ SEACHANGECHECK DELAY	Identifies the delay before checking if a video was not deleted by SeaChange just after a restore service. The default value is 1000.	`DIVAACTOR_ SEACHANGECHECKDELAY= 1000`

SeaChange uses a flat file system. You must specify the parameters as shown in the following table when archiving a clip.

Table C–15   SeaChange BMS and BMC Source/Destination Use

`DIVA_archiveObject` Parameter	Value	Example
FilePathRoot	Leave this field empty	
FileNames	Enter the name of the clip in this field.	`POKEMON`

SeaChange BML Servers

The SeaChange BML (Broadcast Media Library) is a large storage system for SeaChange Archive Format (SAF) files and is based on the RAID[2] technology of the SeaChange BMC platform.

A SeaChange BMC (Broadcast Media Cluster) is a cluster of video servers providing unified storage based on SeaChange RAID[2] technology. Each server of the BMC can deliver files stored on RAID[2] to DIVArchive using the FTP protocol.

DIVArchive uses the FTP protocol to communicate with either a BMS or BMC. You can only overwrite the files when the Actor service is stopped. The file transfer format is SAF (SeaChange Archive Format) only.

> **Note:**   The SeaChange FTP servers do not support directories. All files must be listed under the FTP root directory.

The Automatic Load Balancing feature as described for BMC also exists for BML and operates in a similar fashion.

Table C–16   SeaChange BML Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the BML Node.	`10.80.114.26`
	You can disable the SeaChange Automatic Load Balancing by placing a `$` in front of the IP address of all BMC nodes. The syntax for this is `$IP_Address`.	`$10.80.114.26`
Source Type	SEACHANGE_BML	**`SEACHANGE_BML`**
DIVAACTOR_SEACHANGECHECKDELAY	Identifies the delay before checking if a video was not deleted by SeaChange just after a restore service. The default value is 1000.	*`DIVAACTOR_ SEACHANGECHECKDELAY= 1000`*
DIRECTORY_SERVER_ ENABLED	Identifies whether the BML directory server is enabled or disabled.	Valid values are 1 (enabled) and 0 (disabled). The default value is 1 (enabled).

SeaChange BML clip storage is flat. You must specify the parameters as follows when archiving a clip:

Table C–17   SeaChange BML Source/Destination Use

`DIVA_archiveObject` Parameter	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	Enter the name of the clip in this field.	`OFFICESPACE`

SeaChange BMLe and BMLex Servers

The SeaChange BMLe is the storage subsystem of the latest SeaChange MediaClient architecture. SeaChange BMLe is superseded by the BMLex series.

Both the BMLe and BMLex servers are based on the BML architecture. However Infiniband is used for the cluster interconnect rather than the earlier IOP interfaces. Each node of the cluster is equipped with four FSI ports to provide high speed transfers to and from the BMLe and BMLex.

DIVArchive uses CIFS or FTP protocols to communicate with BMLe and BMLex.

File transfer format is the native format of the files stored on the BMLe and BMLex. Each asset consists of:

MPEG2 Files
MPEG essence, private data (`.pd`) and video index (`.vix`) files.

MXF Files
MXF file (`.mxf`), private data (`.pd`) and video index (`.vix`) if the MXF essence is MPEG2.

When the clip consists of three files (that is, the essence file, `.vix`, and `.pd`), the files are always archived and restored by DIVArchive in the following sequence:

.pd
This is the private data file and the first file archived or restored.

.vix
This is the index file and the second file archived or restored.

Essence File
There is no extension on this file and it is the last one archived or restored.

DIVArchive can restore SAF (SeaChange Archive Format) files from the archive to the BMLe or BMLex. When a SAF clip is restored to a BMLe or BMLex, the SAF file is automatically unwrapped by DIVArchive and the three files are restored to BMLe or BMLex (that is, the essence file, .pd file, and .vix file). This Source/Destination can also restore SAF files from an archived SAF Object to BMLe.

This feature is transparent to you because DIVArchive automatically detects SAF and unwraps it in real time. When a SAF clip is restored to the BMLe, the SAF file is unwrapped by DIVArchive and the name of each file is extracted from the SAF file header. The content is restored to BMLe in the separate files previously described.

BMLe and BMLex generated files support SAF releases SAF 0.1, SAF 1.0, and SAF. SAF may contain two consecutive private data files including a 12 byte .pd file, and a 28 byte .pd file. In this case, DIVArchive will only restore the 28 byte file while ignoring the 12 byte file.

You must declare one Source/Destination for each FSI of each node:

Table C–18   SeaChange BMLe and BMLex Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address FSI	`10.80.114.26`
Source Type	**SEACHANGE_BML**	**SEACHANGE_BML**

Connect Options	`-ftp` or `-cifs`
	`-login {FTP_user_name}`
	`-user {cifs_user_name@domain}`
	`-pass {password}`
	`-nometadata`
		`-cifs`
		`-user me@ourdomain.com`
		`-pass Pa$$word`

DIVAACTOR_ SEACHANGECHECK DELAY	Identifies the delay before checking if a video was not deleted by SeaChange just after a restore service. The default value is 1000.	`DIVAACTOR_` `SEACHANGECHECKDELAY=1000`

-ftp or -cifs
One of these two options must be specified. Otherwise, Streaming API protocol is assumed, which is not supported by DIVArchive for BMLe and BMLex. This option cannot be superseded by the request option.

-ftp
FTP protocol is used for data transfer to and from BMLe and BMLex.

-cifs
CIFS protocol is used for data transfer to and from the BMLe and BMLex FSI cards. The implicit CIFS path to BMLe is \\fsi_ip_address\vstrm.

`-nometadata`

This option prevents DIVArchive from archiving the `.vix` and `.pd` files when the clip being transferred includes essence, `.vix`, and `.pd` files. This option cannot be superseded by the request option.

You must specify the parameters as follows when archiving a clip:

Table C–19   SeaChange BMLe and BMLex Source/Destination Use

`DIVA_archiveObject` **Parameter**	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	Enter the name of the clip in this field.	`HANNITY`

Leitch vR Series Servers

The Leitch vR series video server is connected to external storage that is usually shared with other video servers of the same brand. Clips are stored on Leitch storage as flat files, one file per clip, without any folder structure.

To move clips in and out of the shared storage, Leitch provides a dedicated gateway called the Archive Streamer. The Archive Streamer offers standard FTP protocol over a Gigabit Ethernet network.

> **Note:** The Leitch vR *Source Type* is depreciated. It was initially created to follow the first Archive Streamer releases that did not correctly report the size of the file to be transferred.

You must create one record for each Archive Streamer DIVArchive must move data to and from.

Table C–20   Leitch vR Series Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of Leitch Archive Streamer	`10.80.114.21`
Source Type	**FTP_STANDARD**	`FTP_STANDARD`
Connect Options	`-login {FTP_user_name}` `-pass {FTP_password}` `-port {FTP_port}`	`-login ftpuser` `-pass Pa$$word` `-port 6021`

You must specify the parameters as follows when archiving a clip:

Table C–21   Leitch vR Series Source/Destination Use

`DIVA_archiveObject` **Parameter**	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	Enter the name of the clip in this field.	`FRIENDS`

Leitch Nexio Servers

The Leitch Nexio video server is connected to external storage that is usually shared with other video servers of the same brand. Clips are stored on Leitch storage as flat files, one file per clip, without any folder structure.

To move clips in and out of the shared storage is possible directly from the video server using the standard FTP protocol over a Gigabit Ethernet network.

> **Note:** The Leitch Nexio *Source Type* is deprecated.

You must create one record for each video server DIVArchive must move data to and from.

Table C–22   Leitch Nexio Series Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of Leitch Nexio video server.	`10.80.114.21`
Source Type	**FTP_STANDARD**	**`FTP_STANDARD`**

Connect Options	`-login {FTP_user_name}`
	`-pass {FTP_password}`
	`-port {FTP_port}`
		`-login ftpuser`
		`-pass Pa$$word`
		`-port 6021`

You must specify the parameters as follows when archiving a clip:

Table C–23   Leitch Nexio Series Source/Destination Use

`DIVA_archiveObject` **Parameter**	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	Enter the name of the clip in this field.	`ENEMIES`

Grass Valley Profile Servers

Grass Valley Profile video servers are provided in one of two ways; with independent storage, where the video server includes its own fault tolerant disk storage, or as part of a MAN, where video servers are connected to a SAN where the fault tolerant disk storage resides.

Irrespective of the storage mechanism, the DIVArchive Actor always connects to a specific Profile server. The exchange format is GXF only.

Profile Storage consists of one master disk (for example, `EXT:` or `INT1:`), and one level of folders where one clip is seen as one file. One folder called `default` always exists.

The network infrastructure between GVG Profiles and DIVArchive Actors is an IP/FC network.

You must create one record for each video server DIVArchive must move data to and from.

Table C–24   Grass Valley Profile Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the video server.	`10.80.114.21`
Source Type	**PDR**	**`PDR`**
Name	Logical name for the video server.	`GVG-Profile-1`

The Actor configuration parameters are located in the *Actor* frame of the DIVArchive Configuration Utility. The two parameters in the following table directly influence

transfer performance. Oracle recommends trying several value combinations on the target platform.

In addition to these two parameters, the MTU size setting for the HBA used for IP/FC traffic to the Profile servers may also have an influence on transfer performance.

Grass Valley does not provide any recommendation for MTU size. However, Oracle recommends setting the MTU size on the Actor HBA to the same value as the MTU size of the Profile HBA. *This is only a recommended setting and not an absolute rule.*

Table C–25   Grass Valley Profile Actor Attributes

Attribute	Description	Recommended Values
DIVAACTOR_ PROFILEREADINGBS	The FTP block size (in bytes) used for transfers on Profile video servers in reading.	`1500` `16374` `32768` (default)
DIVAACTOR_ PROFILEWRITINGBS	FTP block size (in bytes) used for transfers on profile video servers in writing.	`16374` `32768` (default)

You must specify the parameters as described in the following table when archiving a clip:

Table C–26   Grass Valley Profile Source/Destination Use

`DIVA_archiveObject` Parameter	Value	Example
FilesPathRoot	`/explodedFile/disk:/folder`	`/explodedFile/INT1:/default`
FileNames	Enter the name of the clip in this field.	`MyClip`

Grass Valley UIM Gateway

UIM is a gateway to standalone or MAN Grass Valley Profile servers. It provides TCP/IP over Gigabit Ethernet connections to external systems (such as DIVArchive). For legacy purposes, the connection can also be IP/FC for regular profiles.

UIM also provides real-time format conversion (to MXF). The UIM exchange format is GXF (by default), or alternately MXF.

You must create one record for each UIM DIVArchive has to move data to and from.

Table C–27   Grass Valley UIM Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address for the UIM.	10.80.114.21
Source Type	**PDR**	`PDR`
Connect Options	`-login {movie|mxfmovie}` `-format {?D10AES3}` `-extension {file_extension}`	`-login mxfmovie` `-format ?D10AES3` `-extension .mxf`

`-login`
Specifies the FTP user for logging onto the UIM to achieve transfers in the desired format. The two available logins are `movie` (for GXF exchange format), and `mxfmovie` (for MXF exchange format). The `movie` user is assumed if `-login` is not specified.

-format

The UIM supported options for some file formats. This depends on `-login` option. The only available option is `?D10AES3`. The `?D10AES3` option is an e-VTR compliant file format used with the `-login mxfmovie` option. If this option is not specified, MXF files will be processed in Grass Valley OP1a format. This option is not specified by default.

This option can be superseded by the request option.

-extension

This option adds the specified extension to the original clip name in the archive. For example, if the original clip is `clip1` and the `-extension .mxf` option is specified, then the archived file will be `clip1.mxf`.

You must suppress the specified extension before restoring to the destination if it already exists. For example, if the archived file is `clip1.mxf` and `-extension .mxf` option is specified, the restored file on the destination will be `clip1`.

This option is deprecated and replaced by the `-arch_renaming` and the `-rest_renaming` options. This option can be superseded by the request option.

UIM are gateways to the Profile server. You use this the same way for UIM and Profile servers regardless of the transfer format (GXF or MXF).

Table C–28   Grass Valley UIM Source/Destination Use

DIVA_archiveObject Parameter	Value	Example
FilesPathRoot	/explodedFile/disk:/folder	/explodedFile/INT1:/default
FileNames	Enter the name of the clip in this field.	MyClip

Grass Valley K2 Servers

From DIVArchive's standpoint, K2 servers are similar to Profiles and UIM combined. K2 servers offer Gigabit Ethernet connections to external systems, and the exchange format is GXF (default), and alternately MXF.

You must create one record for each K2 server DIVArchive must move data to and from.

Table C–29   Grass Valley K2 Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the K2 server.	10.80.114.21
Source Type	**PDR**	**PDR**
Connect Options	-k2 -login {movie|mxfmovie} -format {?D10AES3} -extension {file_extension}	-k2 -login mxfmovie -format ?D10AES3 -extension .mxf

-k2

This specifies the interface with the K2 servers. When this option is set, DIVArchive will retrieve the size of the file to be transferred before the actual archive transfer (K2 FTP does support the `SIZE` command). Correct transfer progress is reported by DIVArchive.

When this option is not set, DIVArchive will assume that servers are Profile, and will not retrieve the file size before archive transfers. Progress will then remain at 0% before suddenly jumping to 100% when the transfer is complete.

This option has no impact on transferred content, and can be superseded by the request option.

-login
This option specifies the FTP user for logging onto the K2 Server to achieve transfers in the desired format. The two available logins are `movie` (for GXF exchange format), and `mxfmovie` (for MXF exchange format). The `movie` user is assumed if `-login` is not specified.

-format
The K2 supported options for some file formats. This depends on `-login` option. The only available option is `?D10AES3`. The `?D10AES3` option is an e-VTR compliant file format used with the `-login mxfmovie` option. If this option is not specified, MXF files will be processed in Grass Valley OP1a format. This option is not specified by default.

This option can be superseded by the request option.

-extension
This option adds the specified extension to the original clip name in the archive. For example, if the original clip is `clip1` and the `-extension .mxf` option is specified, then the archived file will be `clip1.mxf`.

You must suppress the specified extension before restoring to the destination if it already exists. For example, if the archived file is `clip1.mxf` and `-extension .mxf` option is specified, the restored file on the destination will be `clip1`.

This option is deprecated and replaced by the `-arch_renaming` and the `-rest_renaming` options. This option can be superseded by the request option.

You use this the same way for K2 and Profile servers regardless of the transfer format (GXF or MXF).

Table C–30   Grass Valley K2 Source/Destination Use

DIVA_archiveObject Parameter	Value	Example
FilesPathRoot	/explodedFile/disk:/folder	/explodedFile/INT1:/default
FileNames	Enter the name of the clip in this field.	MyClip

Grass Valley M-Series iVDR Servers

Grass Valley iVDR is an analog and digital VTR that includes a Gigabit connection for material exchange of GXF files. The iVDR exchange protocol is similar to the exchange protocol for Profile servers.

you must create one record for each video server DIVArchive must move data to and from.

Table C–31   Grass Valley M-Series iVDR Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the iVDR.	10.80.114.21
Source Type	**PDR**	**PDR**

Table C–31   (Cont.) Grass Valley M-Series iVDR Source/Destination Attributes

Attribute	Value	Example
Name	Logical name for the iVDR.	GVG-iVDR

You must specify the parameters as follows when archiving a clip:

Table C–32   Grass Valley M-Series iVDR Source/Destination Use

`DIVA_archiveObject` Parameter	Value	Example
FilesPathRoot	/explodedFile/disk:/folder	/explodedFile/INT1:/default
FileNames	Enter the name of the clip in this field.	MyClip

Sony MAV70 Servers

The Sony MAV70 video server has its own independent storage. MAV70 storage organization is flat and all files reside in the storage root. A Linux computer in front of each MAV70 provides a standard FTP connection for moving data to and from the video server over a Gigabit Ethernet Network.

You must create one record for each MAV70 server DIVArchive must move data to and from.

Table C–33   Sony MAV70 Source/Destination Attributes

Attributes	Value	Example
IP Address	IP address of the MAV70 server.	10.80.114.21
Source Type	**FTP_STANDARD**	**FTP_STANDARD**
Connect Options	-login {user_name} -pass {password}	**-login wing** -pass mpegworld

You must specify the parameters as follows when archiving a clip:

Table C–34   Sony MAV70 Source/Destination Use

`DIVA_archiveObject` Parameter	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	Enter the name of the clip in this field.	MyClipName

Omneon Spectrum MediaDirector Servers (QuickTime)

The Omneon MediaDirector is the heart of the Omneon Spectrum architecture. It is connected to MediaPorts or MultiPorts which handle isochronous ingest and playback, and to external storage that is usually shared with other Omneon MediaDirectors.

You can use either MediaStore or MediaGrid for external storage. This section describes connecting MediaDirector to MediaStore storage for MediaGrid support in DIVArchive.

Note: MediaGrid is not supported in the Linux environment.

DIVArchive interfaces with an Omneon MediaDirector to move clips in and out of the shared storage, using standard FTP protocol, over a Gigabit Ethernet Network.

When Omneon Spectrum Servers are configured to ingest material in QuickTime format, essence files are stored in a specific folder structure. The DIVArchive Actors use unique FTP site commands for smart and transparent access to essence files (in particular, the automatic discovery of a folders structure and collision-avoidance at restore time).

You must create one record for each MediaDirector DIVArchive must move data to and from.

Table C–35   Omneon Spectrum MediaDirector Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of Omneon Director.	`10.80.114.21`
Source Type	**OMNEON**	`OMNEON`
Root Path	Either leave this field empty or enter an absolute clip directory.	`/default/clip.dir`

Connect Options	`-streaming_mode`
	
	`-sm`
	
	`-tempdir_mode`
		`-streaming_mode`
		
		`-sm`
		
		`-tempdir_mode`

`-streaming_mode` or `-sm`
This option is QuickTime specific and has no effect on the MXF content. If this option is set, DIVArchive will restore the QuickTime reference file in the following sequence:

1. Audio Tracks

2. QuickTime File

3. Video track

The restore workflow is specific when this option is set. DIVArchive uses the temporary folder to cache the QuickTime file.

`-tempdir_mode`
This option performs a Partial File Restore of MXF files, and is applicable only to Omneon servers. The MXF Partial File Restore request will terminate if this option is not included in the request.

Table C–36   Omneon Spectrum MediaDirector Source/Destination Archive Parameters

`DIVA_archiveObject` Parameter	Value	Example
FilesPathRoot	Enter the absolute clip directory in this field, or leave this field empty to use the configured *Root Path*.	`/default/clip.dir`
FileNames	Enter the name of the clip in this field.	`MyClip`

Table C–37   Omneon Spectrum MediaDirector Source/Destination Restore Parameters

DIVA_restoreObject Parameter	Value	Example
FilesPathRoot	Enter the absolute clip directory in this field, or you can leave this field empty to use the configured *Root Path*.	/default/clip.dir

Omneon MediaGrid Content Storage System

MediaGrid is the Content Storage System from Omneon to which Omneon Spectrum servers can be connected.

> **Note:**   MediaGrid is not supported in the Linux environment.

The MediaGrid system consists of two major components; *ContentServers* that store and provide access to media, and *ContentDirectors* that act as overall file system controllers. ContentDirectors manage the distribution of data throughout the system.

Like any other client system, DIVArchive gets access to the media through a MediaGrid ContentDirector. DIVArchive interfaces with MediaGrid using the CIFS protocol exclusively over a Gigabit Ethernet Network.

The MediaGrid ContentDirector manages data access while the data transfer occurs directly to/from the ContentServers. The Omneon File System Driver (FSD), installed on MediaGrid clients hides this complexity to client systems.

> **Note:**   The Omneon FSD must be installed on each Actor exchanging assets with MediaGrid.

The latest release of Omneon FSD for Windows is available for download at http://support.omneon.com/Updates/Omneon/Current/MediaGrid/WinFSD. The password for the site (if required) is alloyparka.

When material is wrapped in QuickTime format, the essence files are stored using a specific folder structure. DIVArchive also uses unique FTP site commands for smart and transparent access to the essence files (in particular, automatic discovery of folders structure and collision-avoidance at restore time).

When the Actor is running as a Windows service, MediaGrid shares are accessed through a UNC path because drive letters mapped to network drives are not accessible by Windows services. In this case ensure the following:

- Omneon MediaGrid folders being accessed by DIVArchive are properly shared for a given Windows user.

- The DIVArchive Actor service is configured to run under this user account.

- The user has local administrative rights on the DIVArchive Actor.

You must create one record for each ContentDirector DIVArchive must move data to and from.

Table C–38   Omneon MediaGrid ContentDirector Attributes

Attribute	Value	Example
IP Address	Leave this field empty.	

Table C–38 (Cont.) Omneon MediaGrid ContentDirector Attributes

Attribute	Value	Example
Source Type	**MEDIAGRID**	`MEDIAGRID`

Root Path	`\\ContentDirector\filesystem\clip.dir`	`\\10.30.0.200\cldev4\clip.dir`
		
		`\\mycontentdir\fs5\clip.dir`

Table C–39 Omneon MediaGrid Source/Destination Use

`DIVA_archiveObject` **Parameter**	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	enter the name of the clip in this field.	`MyClip`

In cases where the asset is wrapped as QuickTime, DIVArchive searches for files matching the format `clipname.mov` or `clipname.MOV`. DIVArchive automatically retrieves and processes all of the potentially referenced files.

In cases where the material is wrapped as MXF, DIVArchive will search for a file matching the format `clipname.mxf` or `clipname.MXF`. There is only one file per clip.

Quantel Power Portal Gateway

The Quantel Power Portal was previously called the *ISA Gateway*. An ISA system consists of the following components:

ISA Manager
The ISA Manager contains the Clip Database. Clips are identified using a unique FID (File Identifier) in the ISA System.

Q or sQ Servers
One or more Q or sQ servers. These servers contain video cards and disk arrays. Each disk array is associated to a POOL ID, and a single sQ Server can have several POOL IDs. For example, sQ Server ID 1 contains POOL ID 1 and POOL ID 2, sQ Server ID 2 contains POOL ID 3, and sQ Server 3 contains POOL ID 4.

ISA Gateway (Power Portal)
This gateway is a FTP server that imports and exports clips.

You must create one record for each Power Portal (ISA Gateway)

Table C–40 Quantel Power Portal Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the video server.	`10.80.114.21`
Source Type	**QUANTEL_ISA**	`QUANTEL_ISA`

Connect Options	`-login {FTP_user_name}`
	
	`-pass {FTP_password}`
		`-login ftpuser`
		
		`-pass Pa$$word`

The Actor configuration parameters are located in the *Actor* frame of the DIVArchive Configuration Utility.

Table C–41   Quantel Power Portal Actor Configuration Parameters

Parameter	Description	Suggested Values
DIVAACTOR_QUANTELRENAMECLIPS	Enables and disables the file renaming feature.	`0` indicates the renaming feature is disabled. `1` indicates the renaming feature is enabled.

DIVAACTOR_QUANTELRENAMECLIPS applies to Restore requests only. If this parameter is set to `1`, and the **Object Name** format is `clipName,UID` (this is Omnibus naming), then object related files are renamed using `clipName` as the **Name Root**.

For example, if the object `Superman,01AB45` is composed of files `8152.D10` and `8152.WAV`, and is restored to a **QUANTEL_ISA** destination, the following is true:

- If *DIVAACTOR_QUANTELRENAMECLIPS* is set to `0` (disabled), DIVArchive transfers files called `8152.D10` and `8152.WAV` to Power Portal.

- If *DIVAACTOR_QUANTELRENAMECLIPS* is set to `1` (enabled), DIVArchive transfers files called `Superman.D10` and `Superman.WAV` to Power Portal.

Quantel storage is a flat structure. You must specify the parameters as follows when archiving a clip:

Table C–42   Quantel Power Portal Source/Destination Archive Use

`DIVA_archiveObject` **Parameter**	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	`FID1.ext1[,FID1.ext2,]` and so on.	`clip.mxf,clip1.tar`

Files coming from Power Portal can be different file types including: D10+WAV (file names similar to `8152.D10` and `8152.WAV`), MXF (`TestClip.mxf`), and TAR (`FramesDifference.tar`).

If a file is restored twice to Power Portal, the first file is not overwritten. The second restore creates a new file that is identified by a new FID. The DIVArchive Actor captures the new FID after the transfer and forwards it to the DIVArchive Manager.

You must call DIVA_GetRequestInfo to obtain the new FID using the DIVArchive API. If the request is completed, the new FID is in the request's `ADDITIONAL_INFO` field within `ClipID` tags. The `ClipID` tag is encapsulated in the `ADDITIONAL_INFO` tag.

```
<ADDITIONAL_INFO>
 <ClipID>8546</ClipID>
</ADDITIONAL_INFO>
```

Automation is also free to specify a *POOL ID* in the **FilePathRoot** Restore request parameter. If no *POOL ID* is specified, Power Portal will automatically assign one at restore time.

Sony Hyper Agent Servers

Hyper Agent is the name given to Newsbase's FTP server from Sony. The implementation of this FTP server is specific because the `LIST` command returns a proprietary formatted list of files. This list contains duration, and start and end time codes, but not the size of the file in bytes. The size of each clip is calculated by the Actor using three values; *duration*, *frame rate* and *bitrate*. The resultant size is not

accurate, but it is enough for the Manager to allocate a tape for all Archive requests. The progress bar is not affected by the approximated size.

Duration, *frame rate* and *bitrate* are retrieved using the following two commands, which are set by the Actor at the beginning of each Archive request:

SITE FSIZ {Clip ID}
This SITE command returns the duration of the specified clip.

SITE GCNF
This SITE command returns the current system configuration of the server. *This system configuration must remain the same to ensure that all of the clips on the server are the same.*

The following example is a log entry of communications between Actor and the Hyper Agent FTP:

> **Note:** In the following log example the word *configuration* is misspelled; this is a bug in the FTP server and appears in logs as shown in the example.

SITE FSIZ 1444247

`200 150` (the *duration* is 150 frames)

SITE GCNF

`213-System configureation`

`PAL` (the *frame rate* is 25 frames per second)

`20`

`30.0` (the *Bitrate* is 30 Mbps)

`D10`

`SD_IFRAMEONLY`

`213 End of system configuraion`

You must create one record for each ClipBox DIVArchive must moved data to and from.

Table C–43   Sony Hyper Agent Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the Newsbase server.	`10.80.114.21`
Source Type	**SONY_HYPER_AGENT**	**SONY_HYPER_AGENT**
Connect Options	`-login {user_name}` `-pass {password}`	`-login sony` `-pass sony`

Table C–44   Sony Hyper Agent Source/Destination Use

DIVA_archiveObject Parameter	Value	Example
FilesPathRoot	Leave this field empty.	
FileNames	Enter the Clip ID in this field.	`1444247`

Standard FTP and SFTP Servers

DIVArchive running in a Windows environment can interface with any standard FTP server (Linux or Windows), and SFTP servers (known as SSH FTP or Secure FTP). *Oracle only supports Linux-based FTP servers when operating in a Linux environment*. The Windows-based FileZilla and IIS FTP servers are not supported in Linux because these servers are incapable of handling large numbers of files.

Video servers supporting a fully RFC-959 compliant FTP server are considered standard FTP servers. The only restriction that applies is that Linux-style directory listings are required. You set this parameter in the *Home Directory* section of the FTP Site Properties for Microsoft IIS FTP servers.

You must create one record for each video server DIVArchive must transfer data to and from.

Table C–45   Standard FTP and SFTP Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the FTP server.	`10.80.114.21`
Source Type	**FTP_STANDARD** or **SFTP**	`FTP_STANDARD`
Connect Options	`-login {user_name}` `-pass {password}` `-port {port_number}`	`-login moon` `-pass mars` `-port 27`

-login
This is the FTP or SFTP user name. The default value is `anonymous`.

-pass
This is the FTP or SFTP user's associated password. The default value is `anonymous`.

-port
This is the port number the FTP or SFTP server is listening on for connections. The default value for **FTP_STANDARD** is `21`, and for **SFTP** is `22`.

You can specify parameters three different ways for Archive requests as described in the following table:

Table C–46   Standard FTP and SFTP Servers Source/Destination Use

`DIVA_archiveObject` Parameter	Value	Example
FilesPathRoot	Full path to files	`/my_videos/movies`
	Partial path to files	`/my_videos`
	No path entry	
FileNames	Names of files	`maniolia, matrix`
	Partial path and names of files	`movies/maniolia, movies/matrix`
	Full path and names of files	`/my_videos/movies/maniolia, /my_videos/movies/matrix`

DISK_FTP_PASSIVE_MODE
By default, data connections are created in active mode. In active mode, the DivaFtp client connects from a random, unprivileged port that is higher than port 1023. Then, it

starts listening on the port and sends a `PORT` command to the FTP server. Valid values for this parameter are `0` (disabled) and `1` (enabled).

When you set `DISK_FTP_PASSIVE_MODE` to `1` (enabled), data connections are created in passive mode. In passive mode, DivaFtp sends a `PASV` command and the server (not the client) creates the socket.

DISK_FTP_BLOCK_SIZE

The `DISK_FTP_BLOCK_SIZE` parameter defines how much data Actor tries to send and receive with a single system call during FTP transfers. For example, if the internal buffer size of Actor is set to 2 MB and `DISK_FTP_BLOCK_SIZE` is set to 32768 bytes, 64 system calls are required to write a single buffer to a data socket. The default value is 32768 bytes.

DISK_FTP_SOCKET_WINDOW_SIZE

The `DISK_FTP_SOCKET_WINDOW_SIZE` parameter adjusts the normal buffer sizes allocated for output and input buffers. `DISK_FTP_SOCKET_WINDOW_SIZE` is internally used to set `SO_SNDBUF` and `SO_RCVBUF` for FTP managed disk types. The default value is 65536 bytes.

Local Sources

A local source represents a disk partition for a specific Actor (internal disks, NAS or SAN disks), and is tied to a specific Actor (versus a disk source not tied to any particular Actor).

You must create one record for each local source DIVArchive must transfer data to and from.

Table C–47   Local Source/Destination Attributes

Attribute	Value	Example
Name	Enter the same name as the Actor this source is bound to.	`actor1`
IP Address	Enter the same IP address as the Actor this source is bound to.	`10.80.114.21`
Source Type	**LOCAL**	**`LOCAL`**

You can specify parameters three different ways for Archive requests as described in the following table:

Table C–48   Local Source/Destination Use

`DIVA_archiveObject` **Parameter**	Value	Example
FilesPathRoot	Full path to files	`/my_videos/movies`
	Partial path to files	`/my_videos`
	No path entry	
FileNames	Names of files	`maniolia, matrix`
	Partial path and names of files	`movies/maniolia, movies/matrix`
	Full path and names of files	`/my_videos/movies/maniolia, /my_videos/movies/matrix`

If NT drive letters (for example `E:`) are used, Oracle highly recommends leaving them in the *FilesPathRoot* section (that is, use scheme 1 or 2 in the previous table).

Including them in the *FileNames* section prevents the request from replacing them with another path at restore time. Therefore, you cannot restore these objects on a different platform (for example a Linux-based FTP server) where drive letters are not considered valid paths.

Disk and CIFS Sources

A **DISK** or **CIFS** source represents a disk partition assumed to be visible from all Production System Actors. The only difference between **DISK** and **CIFS** is the way blocks of data are read and written:

- **DISK** instructs Actors to use (Windows) `Direct I/O`.

- **CIFS** instructs Actors to use (Windows) `Buffered I/O`.

- Both **DISK** and **CIFS** sources support UNC paths.

You must create one record for each **DISK** or **CIFS** source DIVArchive must move data from the source to the destination. You can also create a generic source to represent any type of **DISK** or **CIFS** source.

Table C–49   DISK and CIFS Source/Destination Attributes

Attribute	Value	Example
Name	Enter a nickname for the source.	`generic-disk`
IP Address	Enter the IP address for the source.	`10.80.114.21`
Source Type	**DISK** or **CIFS**	**DISK**

You can specify parameters three different ways for Archive requests as described in the following table:

Table C–50   DISK and CIFS Source/Destination Use

`DIVA_archiveObject` Parameter	Value	Example
FilesPathRoot	Full path to files Partial path to files No path entry	`/my_videos/movies` `/my_videos`

FileNames	Names of files
	Partial path and names of files
	Full path and names of files
		`maniolia, matrix`
		`movies/maniolia, movies/matrix`
		`/my_videos/movies/maniolia, /my_videos/movies/matrix`

If NT drive letters (for example `E:`) are used, Oracle highly recommends leaving them in the *FilesPathRoot* section (that is, use scheme 1 or 2 in the previous table). Including them in the *FileNames* section prevents the request from replacing them with another path at restore time. Therefore, you cannot restore these objects on a different platform (for example a Linux-based FTP server) where drive letters are not considered valid paths. *Oracle only supports Linux-based FTP servers when operating in a Linux environment*. The Windows-based FileZilla and IIS FTP servers are not supported in Linux because these servers are incapable of handling large numbers of files.

Metasources

A Metasource is a collection of several (single) Sources of the same type. It is assumed that all Sources of the Metasource are sharing the same online storage. Each Source of the Metasource should be of the same regular type (that is, any type except **METASOURCE**), aka Metasource Base Type. A Metasource provides load-balancing and failover mechanisms across all single sources of the Metasource.

You must create one record for each Metasource DIVArchive must transfer data to and from.

Table C–51   Metasource Source/Destination Attributes

Attribute	Value	Example	Comments
Name	Name for video server's shared storage.	`gvg-man-production`	
IP Address	`server1 [,server2,server3]` and so on	`10.158.1.10,10.2.5.60 ,97.64.52.3`	`server1, server2,server3` must also be defined in the configuration as regular sources of the same type (all types except **METASOURCE**, **LOCAL**, and **DISK** are permitted, for example, **OMNEON**, **PDR**, and so on).
Source Type	**METASOURCE**	`METASOURCE`	
Production System	Must be the same for Metasource and all single sources.		Manager will not start if there is no match.
Site	Either one or the other of the sites from Metasource single sources.		Site specified for Metasource is considered by Manager for resource selection.
Root Path	You can specify a *Root Path* at the Metasource level.		If the Metasource *Root Path* is null, the *Root Path* from the selected single source is considered.

Table C–51 (Cont.) Metasource Source/Destination Attributes

Attribute	Value	Example	Comments
Max Accesses *Max Write Acc.* *Max Read Acc.* *Max Throughput*	Actual value for Metasource does not matter.		The value from the selected single source is considered. You cannot leave these fields empty. Oracle suggests setting traffic regulation parameters to the sum of all single source's respective parameters. Oracle also recommends that you do not make any changes to this parameter while there are active requests being processed because it can lead to request termination.

Connect Options	`-failover_time={time_in_milliseconds}`
	
	`-retry_actor={number_of_retries}`
		`-failover_time=300`
		
		`-retry_actor=3`	

`-failover_time={time_in_milliseconds}`
When you select a single source to process a request and it fails, the single source is temporarily not considered part of the Metasource for 600 milliseconds. You can change this default value using this option. This option cannot be superseded by the request option.

`-retry_actor={number_of_retries}`
You use this option to specify the number of Metasource single sources to be tried for each Actor that can be part of the request processing. The default, when this option is not specified, is 2.

For example, if the Metasource is defined as `sd1, sd2, sd3`, the set of possible Actors is `a1, a2,` and `-retry_actor` is set to `2`, DIVArchive will try a maximum of four combinations; most likely `a0-sd1, a0-sd2, a1-sd3, a1-sd1`.

This option cannot be superseded by the request option.

You can also specify other single source connection options for the Metasource. The following table indicates the effects for each possible option when specified at the Metasource level:

Table C–52 Metasource Source/Destination Connect Options

Connect Option	Considered?	Comments
`qos=`	No	The `qos` value should be the same for all Metasource single sources, otherwise Manager will not start.

*Table C–52   (Cont.) Metasource Source/Destination **Connect Options***

Connect Option	Considered?	Comments
-login	No	Value from selected single source is considered. Applicable to FTP Source/Destinations.
-user	No	Value from selected single source is considered. Applicable to CIFS Source/Destinations.
-pass	No	Value from selected single source is considered.
-port	No	Value from selected single source is considered.
-allow_delete_on_ source	No	Implicitly assumed to be true if all single sources (implicitly or explicitly) allow deleting on Source. Otherwise, assumed to be false.
-arch_renaming	No	Value from selected single source is considered.
-rest_renaming	No	Value from selected single source is considered.
-file_order	No	Value from selected single source is considered.
-tr_archive_format	Yes	Values specified for single sources do not matter.
-tr_restore_format	Yes	Values specified for single sources do not matter.
-tr_names	Yes	Values specified for single sources do not matter.
-rest_metadata	No	Value from selected single source is considered.
-num_actors_retry=	Yes	Values specified for single sources do not matter.
-ftp	No	Value from selected single source is considered.
-cifs	No	Value from selected single source is considered.
-nometadata	No	Value from selected single source is considered.
-format	No	Value from selected single source is considered.
-extension	No	Value from selected single source is considered.
-k2	No	Value from selected single source is considered.

You use a Metasource the same as any source of Metasource Base Type.

There are instances where it is required to delete content, and possibly the parent folder, on a server. To satisfy all possible scenarios there are two options available:

- -r deletes recursively

- -delete_fpr includes deletion of the parent folder

The two options, -r and -delete_fpr, work either separately or together, as described in the following workflow examples:

Table C–53   Examples for the -r and -delete_fpr Options

FilesPathRoot	Files	Options	Result
C:\source\root	*	-r	DIVArchive deletes the content of C:\source\root recursively.
C:\source\root	*	-r -delete_fpr	DIVArchive deletes the content of C:\source\root recursively, and then deletes root.

Table C–53 (Cont.) Examples for the `-r` and `delete_fpr` Options

FilesPathRoot	Files	Options	Result
`C:\source\root`	`*`		DIVArchive deletes the content of `C:\source\root` (first level only).
`C:\source\root`	`*`	`-delete_fpr`	DIVArchive deletes the content of `C:\source\root` (first level only), and then eventually deletes `root` if it is empty.
`C:\source\root`	`obj\*`	`-r`	DIVArchive deletes the content of `C:\source\root\obj` recursively, and then deletes `C:\source\root\obj`.
`C:\source\root`	`obj\*`	`-r -delete_fpr`	DIVArchive deletes the content of `C:\source\root\obj` recursively, then deletes `C:\source\root\obj`, and finally deletes `C:\source\root` if it is empty.
`C:\source\root`	`obj1\*` `obj2\*`	`-r`	DIVArchive deletes the content of `C:\source\root\obj1` recursively, then deletes `C:\source\root\obj1`, and then deletes the content of `C:\source\root\obj2` recursively, and finally deletes `C:\source\root\obj2`.
`C:\source\root`	`obj1\*` `obj2\*`	`-r -delete_fpr`	DIVArchive deletes the content of `C:\source\root\obj1` recursively, then deletes `C:\source\root\obj1`, then deletes the content of `C:\source\root\obj2` recursively, then deletes `C:\source\root\obj2`, and finally deletes `C:\source\root` if it is empty.

Table C–53 (Cont.) Examples for the `-r` and `-delete_fpr` Options

FilesPathRoot	Files	Options	Result

`C:\source\root`	`obj1\*`
	
	`obj2\subfolder\clip.mov`
		`-r -delete_fpr`	DIVArchive deletes the content of `C:\source\root\obj1` recursively, then deletes `C:\source\root\obj1`, then deletes the content of `C:\source\root\obj2\subfolder\clip.mov`, then deletes `C:\source\root\obj2\subfolder` if it is empty, and then deletes `C:\source\root\obj2` if it is empty, and finally deletes `C:\source\root` if it is empty.

Expedat Servers

DIVArchive can interface with DataExpedition Expedat servers, also known as *servedat*. This solution uses MTP, which is a high performance file transfer protocol. This WAN acceleration software can use 100 percent of the bandwidth of any long distance or high latency networks.

See the DataExpedition Expedat Server Installation Manual for detailed information on installation and configuration.

This Source/Destination works similar to the **FTP_STANDARD** Source/Destination in terms of the *FilesPathRoot* and list of files.

When Expedat Server is configured with folders having the *RestrictHome* setting enabled, the *RootPath* for the Data Expedition Source/Destination entry must not reference an absolute path. The *RootPath* may be interpreted as a path that is not accessible from the Expedat home directory. For example, the *Root Path* / is interpreted as `C:\`. However, if the Expedat home directory is `D:\folder`, Expedat will attempt to access the path `D:\folder` on `C:\`, which is not valid. If the home directory is `C:\folder`, using the *Root Path* / is acceptable.

Instead of using an absolute path, relative path addressing must be used to resolve this situation. You accomplish relative path addressing by leaving the *Root Path* field empty in the Configuration Utility, or specifying the relative path in the *FilesPathRoot* field of the GUI Manager or API request for the archive or restore operation.

To set up a default home location so that an API request can always use `""` files path, the Expedat `cv_password.txt` file must contain a log in account assigned to a folder with the *RestrictHome* option set.

For example:

```
diva:diva:::S:\DFM:RestrictHome
diva1:diva:::S:\DFM1:RestrictHome
diva2:diva:::S:\some_other_folder:RestrictHome
```

The separate user log in and password accounts allow for the creation of more than one EXPEDAT Source/Destination entry with different home locations. The API

request can then reference the EXPEDAT Source/Destination pointing to the desired home location.

When you use DFM to monitor an FTP folder in a Linux environment, it must be configured similar to the following example:

User: `diva`

User home directory: `/ifs`

Folder to be Monitored: `/ifs/folder1`

Correct DFM Configuration: `ftp://diva:password@host_ip/folder1`

Incorrect DFM Configuration: `ftp://diva:password@host_ip/ifs/folder1`

You must create one record for each Expedat server DIVArchive must transfer data to and from.

Table C–54   Expedat Server Source/Destination Attributes

Attribute	Value	Example
IP Address	IP address of the Expedat server.	`10.80.114.21`
Source Type	**EXPEDAT**	**EXPEDAT**

Connect Options	`-login {user_name}`
	`-pass {password}`
	`-port {port_number}`
	`-license {license_code}`
	`-encryption`
	`-seq_buffer_size {size_in_megabytes}`
	`-exp_maxrate {size_in_kilobytes}`
	`-exp_mindatagram {size_in_bytes}`
		`-login moon`
		`-pass mars`
		`-port 8080`
		`-license 46FE464A98`
		`-encryption`
		`-seq_buffer_size 16`
		`-exp_maxrate 1024`
		`-exp_mindatagram 2848`

`-login` and `-pass`
These options are mandatory if the server is configured with authentication parameters.

`-port`
This option should always be present because there is no default value.

`-license`
This is a mandatory parameter to use the DIVArchive Expedat Client. Without the license code the EXPEDAT Source/Destination is unusable. You can only configure one Expedat license key per production system.

`-encryption`
This option works with the Expedat Source/Destination, is optional, and enables Expedat content encryption during transfers.

`-seq_buffer_size {size_in_megabytes}`
This option defines the size of the DataExpedition internal buffer for each transfer. The default value is 16 MB and should be sufficient for most transfers. A large buffer allows DataExpedition to continue moving data during times when the sender or receiver may not be able to process it. However, a small buffer consumes less memory.

`-exp_maxrate {size_in_kilobytes}`
This option sets an approximate limit on the number of kilobytes per second, per transfer. The default is unlimited, but can be used as an alternate method of controlling bandwidth.

`-exp_mindatagram {size_in_bytes}`
This transfer protocol is over UDP. This option defines a minimum size for each network datagram payload that DataExpedition sends. The purpose is to prevent DataExpedition from sending too small of a packet over the network. You may want to set this value between 2848 and 8544 when using a very fast network path (Gigabit or higher) and every device along the path supports Jumbo Frames (MTU 9000). Using large datagrams can greatly reduce CPU overhead. However, using this setting without Jumbo Frames being fully supported can cause severe performance issues or loss of connectivity.

D

ADIC SDLC Installation and Configuration

This appendix describes installation and configuration of the SDLC Server and SDLC Client and includes the following information:

- SDLC Server

  - Prerequisites

  - Configuration

- SDLC Client

  - Installation

  - Configuration

- Using `dasadmin` Commands

- Troubleshooting

SDLC Server

The following sections describe prerequisites and configuration of the SDLC Server.

Prerequisites

The SDLC Server process is called *supervisor*. The SDLC GUI is also available as an applet in your web browser address bar. You access the GUI by entering the IP address of the computer on which SDLC Server is running in the browser address bar.

Avoid stopping the SDLC Manager (that is, the *NobleNet PortMapper for TCP* Windows service) while SDLC Clients are currently connected (for example, the SDLC GUI connection). If the service is stopped, the SDLC Server will vary to a transient state making it temporarily impossible to restart.
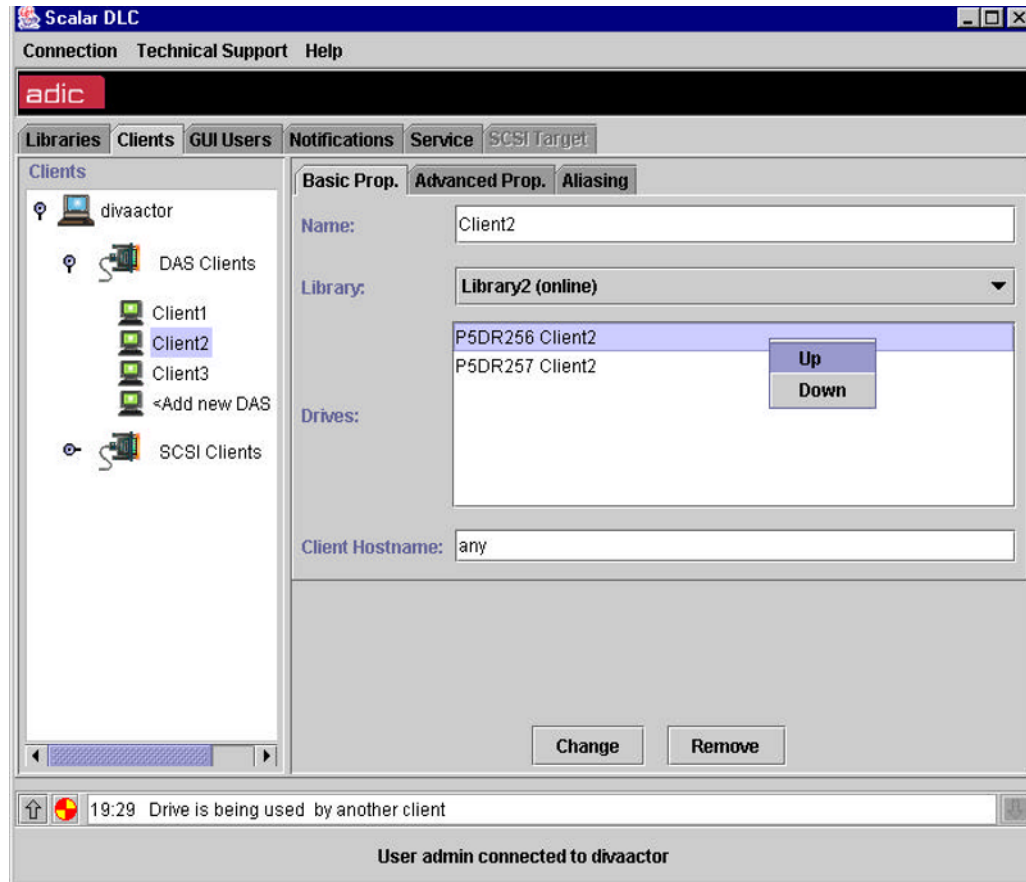
Configuration

You must first define a physical resources partition (in the **SDLC GUI Libraries** tab, then the **Wizard** tab) to make the SDLC usable. After you define the physical resources, you define a logical library with its slots and drives. When the wizard completes, the partition is automatically bound to an ADIC Client. The DIVArchive ADIC Robot Manager uses the client to obtain status information about library items, and to send mount and dismount commands.

Confirm that the drives being used by DIVArchive are bound to the client dedicated to the DIVArchive ADIC Robot Manager. In the following figure, two drives are bound to Client2.

Use the following procedure to bind a drive to a specific client:

1. Open the SDLC GUI.

2. Click the **Clients** tab.

3. Select the client from the *Clients* navigation tree on the left.

4. Right-click the desired drive in the right *Drives* area.

5. Select **Up** from the menu.

   After a drive is bound to a client, the name of the client is appended to the name of the drive.

6. Confirm that for the given client (*Client2* in the previous example), the ***Client Hostname*** field is configured with the host name or IP address of the client computer; that is, the client that will use the *Client2* connection when communicating with the SDLC Server.

   You can specify **any** in this field to accept any incoming connections from any client computer that provides *Client2* as the client name when connecting through the SDLC API. You must use the **any** keyword to use the `dasadmin` tool from a supervising computer.

7. Confirm the *NobleNet PortMapper for TCP* Windows service is started. You must start the service if it is not running.

SDLC Client

The SDLC Client must be installed on the computer where the DIVArchive ADIC
Robot Manager is installed.

Installation

Install the SDLC Client from the SDLC distribution. You are prompted for the name of
the client being used by the ADIC Robot Manager to connect to SDLC Server during
installation. You must use the client you created in the SDLC Server. This is *Client2* in
the example.

> **Note:** The client name is case-sensitive.

Confirm the *NobleNet PortMapper for TCP* Windows service is started. You must start
the service if it is not running.

Configuration

You must define two Windows environmental variables on a Windows system as
follows:

Table D–1   SDLC Windows Environment Variables

Environment Variable	Definition	Example
DAS_SERVER	Host name or IP address of the computer where the SDLC Server has been installed.	10.201.10.100
DAS_CLIENT	Name of the client that the DIVArchive Robot Manager uses to connect to SDLC.	Client2

Use the following procedure to test the SDLC Client connection to the SDLC Server:

1. Open a Windows command line window.

2. Change to the `C:\Program Files\ADIC\SDLC\Bin` folder.

3. Execute `dasadmin qversion`.

   The output will be similar to the following, and then you will be back at the
   command prompt.

   ```
   ACI-Version: 3.10E
   DAS-Version: 3.10
   ```

Using `dasadmin` Commands

The following is a list of commands used when executing the `dasadmin` application.
You must always execute `dasadmin` from the `C:\Program Files\ADIC\SDLC\Bin` folder.

Getting Help
```
dasadmin -h
```

Mounting a Tape
```
dasadmin mount {tape_id} [drive]
```

The `tape_id` is required. If `drive` is not specified, the first free drive is chosen automatically.

Dismounting a Tape

```
dasadmin dism {tape_id}
```

Alternatively you can execute `dasadmin dism {drive_name}`. The `drive_name` is the name of the drive to dismount.

> **Note:** the tape must first be ejected with a SCSI unload before dismounting.

Ejecting a Tape

```
dasadmin eject2 {tape_name} {eject_or_insert_slot_name}
```

> **Note:** Depending on the server configuration, the eject and insert area (that is, slots from the CAP) can have different names.

Inserting a Tape

```
dasadmin insert2 {-n|-c} {eject_or_insert_slot_name}
```

You use the `-n` to specify data tapes and the `-c` to specify cleaning tapes.

> **Note:** Depending on the server configuration, the eject and insert area (that is, slots from the CAP) can have different names.

Querying Drives

```
dasadmin ld
```

Retrieving the Tapes List

```
dasadmin qvolsrange "" "" {number_of_tapes_to_list}
```

Parking the Robot Arm

```
dasadmin robhome {robot_number}
```

Synchronizing the SDLC Database and Library

```
dasadmin inventory
```

Retrieving Tape Information

```
dasadmin view {tape_id}
```

`dasadmin` Release Information

```
dasadmin qversion
```

Library Configuration Information

```
dasadmin eif_conf
```

> **Note:** This command is not supported in SDLC 2.1 and later.

`dasadmin` References

See the `sdlc_doc.pdf` file on the SDLC Installation CD.

Troubleshooting

The `dasadmin qversion` command may not respond as previously stated. The following list identifies the most common cases and remedies:

RPC failure error dialog box appears
A dialog box appears on the screen with the title ACI0004 Function `clnttcp_create` Failed, and the following error displays in the command window:

```
version failed: An RPC failure occurred.
ACI-Version : 3.10E
DAS error = 1
```

To resolve this issue, confirm on the server that you can to connect to this client from the computer where you launched `dasadmin`.

Invalid host name or IP Address error in command window
The following error appears in the command window:

```
version failed: Invalid hostname or IP Address
ACI-Version : 3.10E
DAS error = 14
```

To resolve this issue, confirm on the server that you can to connect to this client from the computer where you launched `dasadmin`. The client host name is probably set to `localhost`.

Invalid pointer to IDAS interface error in command window
The following error appears in the command window:

```
version failed: Invalid pointer to IDAS interface
ACI-Version : 3.10E
DAS error = 28
```

To resolve this issue confirm the `DAS_CLIENT` environment variable is set properly.

The command never ends (endless loop)
If the command results in an endless loop and never stops, confirm the following:

- Confirm the SDLC Server is started.

- Confirm the `DAS_SERVER` environment variable is set properly.

- Confirm the *NobleNet PortMapper for TCP* Windows service is running.