DIVA Core

Operations Guide

Release 8.1

Version 1.2

April 2023

Copyrights and Trademark Notices

Contents

3 Architecture Overview

8 DIVAmigrate Operations

9 Monitoring and Error Handling

Glossary

List of Tables

Preface

This book outlines general operational guidelines for the DIVA Core Suite 8.1. Included are start-up and shut-down procedures for various software and hardware components of a typical DIVA Core system, and the control and monitoring aspects of the DIVA Core platform using the Control GUI and DIVA Command.

Audience

This book is intended for operations and administration personnel.

Documentation Accessibility

For information about Telestream's commitment to accessibility, visit the Telestream Support Portal located at https://www.telestream.net/telestream-support/diva/support.htm.

Access to Telestream Support

Telestream customers that have purchased support have access to electronic support through the Telestream Support Portal located at https://www.telestream.net/telestream-support/diva/support.htm.

Related Documents

For more information, see the DIVA Core documentation set for this release located at https://www.telestream.net/telestream-support/diva/support.htm.

For information on Cloud Storage visit the following links:

Metered and non-metered Oracle Cloud Storage:
http://docs.oracle.com/en/cloud/get-started/subscriptions-cloud/csgsg/

Up to date Oracle Cloud information:
http://docs.oracle.com/cloud/latest/

EMC ECS (*Elastic Cloud Storage*)
https://www.delltechnologies.com/ru-by/learn/data-storage/ecs.htm

Amazon S3 Cloud Storage
https://aws.amazon.com/s3/

Scality Zenko Integration
https://www.zenko.io/what-is-zenko/

Cloudian
https://cloudian.com/

NetApp StorageGrid
https://www.netapp.com/cloud-services/

Alibaba OSS
https://www.alibabacloud.com/product/oss

For further assistance:
https://www.telestream.net/telestream-support/diva/support.htm

Document Updates

The following table identifies updates made to this document.

Date	Update
April 2020	Updated book to include DIVAmigrate and Export / Import operations information.
June 2020	Updated Transcoder section under Config Utility System Tab. Renamed Chapter 7 to Removable Media Operations. Update Amazon S3 bucket limitations.
July 2021	Rebranded book to Telestream Updated copyright notices Updated information for DIVA Core 8.1 release including renaming various components as follows: • *Actor* is now named ***Datahub*** • *Proxy Actor* is now named ***Proxyhub*** • *Production System* is now named ***Network*** • *Source/Destination* is now named ***Server*** Removed redundant terminology Updated information on UNC paths and CIFS running under Linux Updated references to the Configuration Utility with DIVA Command Minor formatting adjustments Removed Backup Service error notifications and warnings (*they are no longer displayed in the Control GUI*) Added DIVA Command information Added Backup Service email notification information Added Backup Service Failover procedures Updated Manager Failover procedures Added Cluster Failover procedures Added Backup Initiator information
September 2021	Included additional links for documentation on various cloud services to the Preface. Minor formatting updates Minor restructuring for better readability
April 2023	Added note referencing not using a dollar sign ($) at the beginning of an object name.

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.
blue text	Blue text indicates a link to an outside source, or to another chapter, section, or glossary term in this book.

1

Overview

This chapter describes the general concepts of a DIVA Core system and includes the following information:

- DIVA Core Overview
- New and Enhanced Features and Functionality

DIVA Core Overview

The DIVA Core architecture enables integration of many different types of servers and technologies (*for example, Broadcast Video Servers, Storage Area Networks, and Enterprise Tape Libraries*). The installation of DIVA Core varies from site to site, so the exact configuration of your specific DIVA Core platform is not covered in this guide. Refer to your System Administrator and your DIVA Core Site Configuration for details on your specific DIVA Core System installation and configuration.

The Site Configuration is a document prepared by Telestream (*or an affiliated system integrator*) after installation of DIVA Core at your site. It is a complete record of the configuration of the DIVA Core sub-systems, third party interfaces, site details and contacts, user names and passwords, and remote access codes.

Your Site Configuration is referenced at various points in this guide. Refer to your Site Configuration where applicable.

New and Enhanced Features and Functionality

Refer to the DIVA Core Release Notes for 8.1 at *https://www.telestream.net/telestream-support/diva/support.htm*.

DIVA Command

DIVA Command replaces the Configuration Utility starting with this release to configure a DIVA Core system. It can be run on any computer that has TCP/IP connectivity to the host running the DIVA Core database.

Caution: DIVA Command is intended only for experienced users. Incorrect or incomplete changes in DIVA Command can adversely affect DIVA Core operations, possibly delete data from the archive, or prevent the Manager from running. If you are unsure about making changes, contact Telestream Support for assistance before attempting to make alterations to the system configuration.

DIVA Command primarily connects to the DIVA Core Database, and for some tasks, directly to the Robot Managers. After launching the utility you must first connect to the database to edit the DIVA Core system configuration. The Oracle username and password for DIVA Core should be documented in the DIVA Core Site Configuration for your site. Although used primarily for configuration of DIVA Core, some operational functions are also performed from DIVA Command.

2 Concepts

This chapter describes the different concepts used in the DIVA Core system and includes the following information:

Source and Destination Servers

A *Source* Server is defined as any connected system that has content intended to be transferred to DIVA Core. A *Destination* Server is defined as any connected system that requires content to be transferred to it from DIVA Core. Examples of both are Broadcast Video Servers, FTP Servers, or Disk Storage.

Datahubs in the Linux operating system support UNC paths for CIFS sources and destinations. The Datahubs will automatically mount the SMB shares to access the Source/Destination Servers.

UNC paths are supported for SMB Servers and managed disks if the UNC path is directly mounted on the Windows Datahubs.

The Source and Destination Servers that are used in DIVA Core requests are predefined in the DIVA Core configuration and are accessible by the **Sources Destinations** button under the **Home** tab. In DIVA Core's Server configuration, each server type or disk file system is given a unique name and are configured as follows:

Source Only

DIVA Core will only archive files from the server or disk file system.

Destination Only

DIVA Core will only restore files to the server or disk file system.

Source and Destination

DIVA Core will archive and restore files to and from the server or disk file system.

Although a detailed explanation of the configuration of a Source or Destination Server is beyond the scope of this guide, a brief overview of the configuration is included because they can influence how requests are issued to them, and influence how two or more simultaneous requests to them are managed in the *Current Requests Queue*.

Generally, each Source and Destination Server has the following parameters configured. These are common to all requests that involve that Server:

- The *Source Type* is the protocol or access method used when interacting with the target device.
- The maximum number of read and write transfer sessions and the total maximum number of read/write sessions combined. This identifies the limits on the number of simultaneous requests that DIVA Core will execute concurrently on the target device, or prioritizing write (*Restore*) operations over read (*Archive*) operations.
- Define the maximum bandwidth allowable to DIVA Core for transfers to or from the device. This may be used to throttle data transfers where the target device is shared with other Networks or third party applications.
- The *Default Quality of Service* (*QOS*). This is the QOS used when **Default** is specified in a request's ***Quality of Service*** field.
- Define *Connect Options* that must be provided (*or that can also be optionally specified*) for the specific protocol or access method of the target device. Examples of *Connect Options* are recursive subfolders, user names or passwords, or other options specific to the selected source type. DIVA Core ignores this parameter if no options are specified.
- The *Root Path* to the files to be archived on the source, or restored to on a destination. This is always specified as an absolute directory path on the target device. For example c:\Exported\MPEG2 for Windows based file systems, or /Movies/MPEG2 on Linux based file systems. The *Root Path* configuration also depends on the source type, and can be left blank in some cases (*and will be ignored by DIVA Core*).

  For **Local** or **Disk** source types, the *Root Path* specifies the mount point of the device in the local file system.

If the ***Connect Options*** and ***Root Path*** parameters have been defined for a Server configuration, they may not be appropriate for every request submitted. DIVA Core allows these parameters to be specified in a DIVA Core request to that source or destination (*at the request level*). Whether a request can override these Server attributes depends on the source type. See the *DIVA Core Source and Destination Servers Table* in the *DIVA Core Installation and Configuration Guide* for a comprehensive list of these options, paths, and how they interact with those specified at the request level.

The ***Files Path Root*** specified in a request can either be appended to the ***Root Path*** specified in the Server configuration, or override the ***Root Path*** entirely if it is specified as an absolute path.

EMC ECS (*Elastic Cloud Storage*) Integration

Instances stored on EMC Elastic Cloud Storage are local instances whose priority is lower than other types of local disk instances, but a higher priority than tape storage instances.

In DIVA Core 8.1 you can define Oracle ***Storage Class*** and ***Storage Location*** separately. If you require new cloud or local arrays in the future, you can specify all of these parameters as options. However, in DIVA Core 8.1 both SWIFT and S3 are supported for interfacing with EMC ECS, but you cannot change the existing configuration after the Array is configured.

You can set the ***Media Priority*** of a source instance for a Restore, Partial File Restore, and Copy to Group requests, which enables restoring an instance stored on a local non-EMC ECS array with a higher priority than an instance on an EMC ECS array. If the priorities for the media are all the same, then the Manager decides which source instance is preferred during these requests.

See the *Installation and Configuration Guide* for information.

OCI (*Oracle Cloud Infrastructure*)

DIVA Core 8.0 (*and later*) includes support for storing your data in Oracle Cloud Infrastructure. The Control GUI is enhanced to support OCI storage operations. OCI services combine cloud elasticity and utility with granular control, security, and predictability of your on-premise infrastructure. OCI delivers high performance, flexibility, availability, and is cost-effective.

Note: If you have a multiple DIVA Core sites, connected to the same OCI / OCI Classic storage account, you must use a different Array Name per site. The Array Name is used to uniquely identify content of an array in the cloud, and therefore must be different. This constraint is not required for other cloud vendors.

Amazon S3 Integration

DIVA Core 8.1 includes support for Amazon S3 AWS integration. Storage accounts allow a user to configure programmatic access to a user's AWS account. The configuration data in a DIVA Core storage account is exclusively used by DIVA Core's Datahubs to query S3 storage and transfer content to and from S3 buckets.

Upon creation of a storage account, DIVA Command will automatically create the set of DIVA Core resources needed to store DIVA Core managed objects in S3. The resources generated by DIVA Command are an Array, Disk, and Datahub-Disk connections.

See the Installation and Configuration Guide for more information.

Scality Zenko

DIVA Core 8.1 includes support for Scality Zenko integration as both Storage Accounts and Servers. Storage accounts allow users to configure programmatic access to a user's account. The configuration data in a DIVA Core storage account is used by the DIVA Core Datahubs to query storage and transfer content. Storage Class supported = STANDARD.

Cloudian

DIVA Core 8.1 includes support for Cloudian integration as both Storage Accounts and Servers. Storage accounts allow users to configure programmatic access to a user's account. The configuration data in a DIVA Core storage account is used by the DIVA Core Datahubs to query storage and transfer content. Storage Class supported = STANDARD.

NetApp StorageGrid

DIVA Core 8.1 includes support for NetApp StorageGrid integration as both Storage Accounts and Servers. Storage accounts allow users to configure programmatic access to a user's account. The configuration data in a DIVA Core storage account is used by the DIVA Core Datahubs to query storage and transfer content. Storage Class supported = STANDARD.

Alibaba OSS

DIVA Core 8.1 includes support for Alibaba OSS integration as both Storage Accounts and Servers. Storage accounts allow users to configure programmatic access to a user's account. The configuration data in a DIVA Core storage account is used by the DIVA Core Datahubs to query storage and transfer content. Storage Class supported = STANDARD. ARCHIVE bucket support is planned for a future DIVA Core release.

Refer to the Installation and Configuration Guide for more details on these storage and Servers.

Data Expedition Integration

DIVA Core can (*optionally*) interface with the Server named **Data Expedition Expedat Server**. The Expedat Server (*also known as **servedat***) is very similar to the **FTP_STANDARD** server and CIFS, and offers AES encryption capabilities. The main difference among them is the protocol used for operations.

The *Expedat Client API* is integrated into the Datahub computer and the Expedat server is integrated into DIVA Core (*either on an Datahub computer or other additional server within the system*) just like the **FTP_STANDARD** server and CIFS, but is faster when used on high latency networks when using the Data Expedition Expedat MTP Protocol (*a high performance file transfer protocol*), which provides better bandwidth utilization.

One record is created for each Expedat Server that DIVA Core has to move data to or from. Although the initial solution for DIVA Connect transfer and restore is still functional in DIVA Core 8.1, the functionality has been enhanced and includes complex objects. With the new functionality, there are only 2 steps required for archiving objects through DIVA Connect instead of 3 steps as previously required.

Source and Destination Server Configuration

One record is created for each Expedat server DIVA Core must move data from or to. *Refer to the Installation and Configuration Guide for more information on Oracle Storage Cloud (OPC and OCI) and EMC integration*. The following are the parameters and examples for Expedat Source and Destination Servers:

IP Address

This is the IP address of the Expedat server.

Example:

10.80.114.21

Source Type

Set this to **EXPEDAT**.

Connection Options

The following are connection options. Some are mandatory while others are optional.

-login *username*

This is mandatory if the server is configured with authentication parameters. For example, -login moon.

-pass ***password***
This is mandatory if the server is configured with authentication parameters. For example, -pass ph4!hi4.

-port *portNumber*
This must be supplied because there is no default value. For example, -port 8080.

-license *licenseCode*
This is mandatory and is the Expedat license number. For example, -license 46FE464A98.

-encryption
This is optional and there are no additional parameters. For example, -encryption.

-seq_buffer_size *megabytes*
Defines the size of the Data Expedition internal buffer for each transfer. The default value is 16 MB and will be sufficient for most transfers. A large buffer allows Data Expedition to continue moving data during times when the sender or receiver may not be able to process it. A small buffer will consume less memory. For example, -seq_buffer_size 16.

-exp_maxrate *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. For example, -exp_maxrate 1024.

-exp_mindatagram *bytes*
This transfer protocol is over UDP. This option can define a minimum size for each network datagram payload that Data Expedition will send. The purpose is to prevent Data Expedition from sending too small of a packet over the network. Set this value between 2848 and 8544 when using a very fast network path (*gigabit or later*) 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. For example, -exp_mindatagram 2848.

Arrays, Disks, and Cache

DIVA Core uses HDD (*Hard Disk Drive*) technologies for both the storage of objects and for transient storage during data transfers (*disk cache*).

Any disk that DIVA Core uses is assigned to an array. An array is a logical association of one or more disks for the storage of objects. Disks that are configured as cache disks are also assigned to an array, typically named CACHE.

The storage of an object on a disk in DIVA Core is identified by the array's name rather than an individual disk itself. DIVA Core automatically allocates objects among two or more disks within any array.

Each disk in any array may be connected to a DIVA Core system either directly in an Datahub host's hardware, as NAS (*Network Attached Storage*), or connected through a SAN (*Storage Area Network)* using Fiber Channel. In the case of SAN, it can also employ additional file system sharing software on the hosts if the disk is to be accessed simultaneously from multiple Datahubs.

Note: See the *Installation and Configuration Guide* for information on configuring EMC ECS components, including Object Stores in arrays, Datahub-Disk connections, Array Priority, Object Storage Accounts, and configuration validation checks.

You can configure individual disks in an array as follows:

Storage Only

The disk will only be used for storage of objects. These types of disks will employ some level of RAID technology to ensure data redundancy and protection against individual hard disk failure.

Storage and Cache

The disk will be used for the storage of objects and also for caching operations. Both types will use separate subfolders on the disk. These types of disks will employ some level of RAID technology to ensure data redundancy and protection against individual hard disk failure.

Cache Only

The disk will only be used for caching, tape to tape copying, tape spanning, or tape repacking operations. These types of disks may employ RAID technology to improve performance (*for example, RAID 0*).

Storage and Nearline

The disk will be used for the storage of objects, and also for Nearline operations. Both types will use the same subfolder on the disk. These types of disks will employ some level of RAID technology to ensure data redundancy and protection against individual hard disk failure.

Cache and Storage and Nearline

The disk will be used for the storage of objects, caching, and Nearline operations. Both storage and Nearline types will use the same subfolder on the disk. However, the cache type will use a separate subfolder. These types of disks will employ some level of RAID technology to ensure data redundancy and protection against individual hard disk failure.

DIVA Core also enables individual disks to be configured for Read-Write access, Read-Only access, or can be disabled temporarily.

The file system of any DIVA Core managed disk should never be manipulated directly by any file manager or utility (*such as Windows Explorer*) or equivalent. If the folder structures or files are moved, renamed or deleted, this may cause DIVA Core to mark that disk as **Out of Order**.

Caution: Using such a utility in any fashion will completely destroy the file system on that disk.

Disks that have file sharing software installed to provide shared host access (*for example, SNFS or MetaSAN*) can appear as an unknown file system or as not initialized to utilities such as Windows Disk Manager.

Disk Discovery

DIVA Core now supports the retrieval of non-complex objects and instance metadata from an OCI/S3 cloud account to a DIVA Core database. This allows the user to update their on-premise DIVA Core system with content from the cloud. A new web-based GUI allows a user to start, resume, or stop a disk-discovery scan and to view the status of a scan. *Refer to the Installation and Configuration Guide for detailed information on how to configure Disk Discovery*.

Tape Groups and Sets

Disks are logically assigned to arrays for the storage of objects, but tapes are logically associated together in *Groups*.

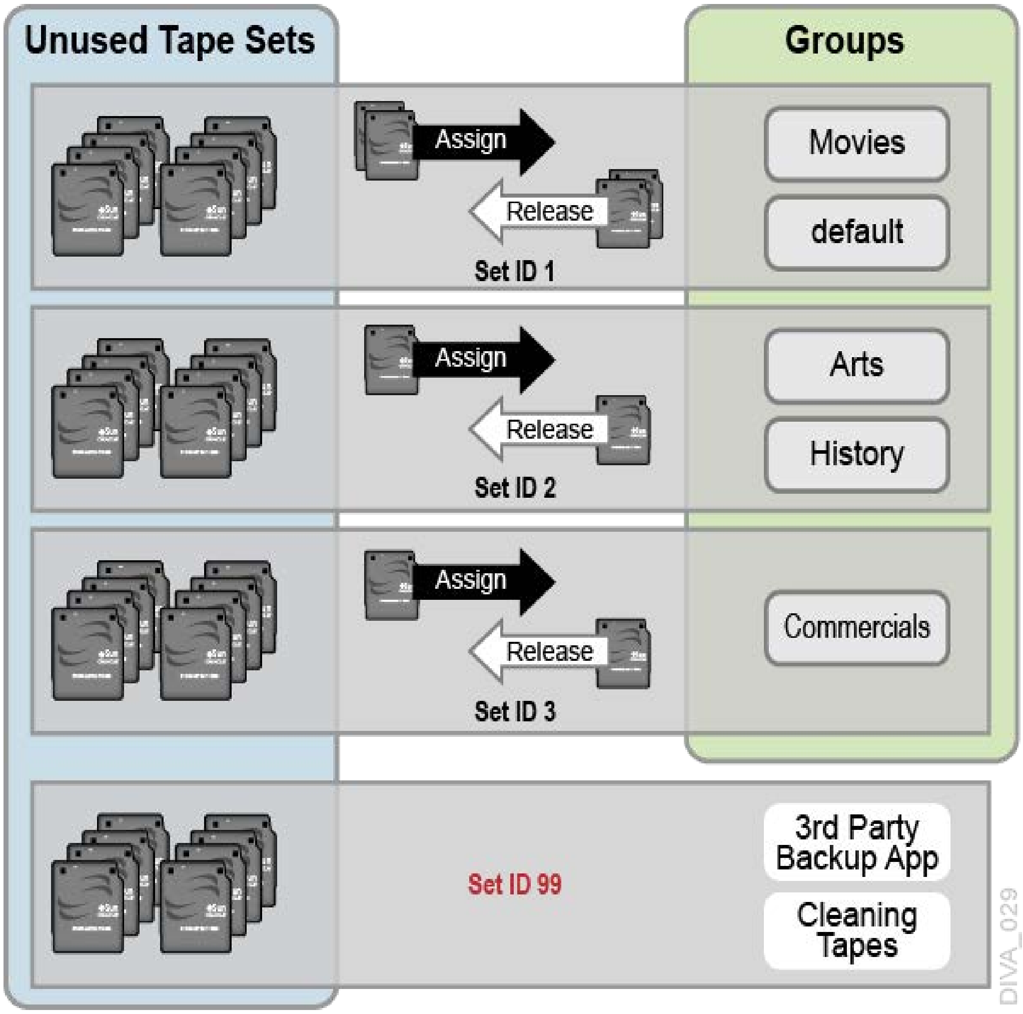
Tapes are initially divided into *Sets*, and assigned a number called a *Set ID*. A Set ID enables partitioning of pools of tapes in a library and assigning them for use with specific groups. A group draws upon the pools by associating the group with a Set ID.

More than one group may use the same Set ID. An unused tape will not actually belong to any of those groups until DIVA Core writes the first object to that tape. When all objects have been deleted from a group-assigned tape, it will be unassigned from that group, and may subsequently be assigned to another group using the same Set ID.

Since groups are a user-defined concept they can differ from one DIVA Core installation to another. The exception is the **Default** group, which exists in all installations and can neither be renamed nor removed. In a DIVA Core system, groups are created and managed in DIVA Command.

When a tape is assigned a Set ID of 99 it indicates to DIVA Core that the tape is not to be used and is not related to the operation of DIVA Core. Examples are tapes that belong to a non-DIVA Core application in a shared library environment or the library's cleaning tapes.

The following figure shows how tape sets and groups are associated and used:



Tape Compression

Tape compression is supported at the tape group level starting with the DIVA Core 8.0 release. You enable (*or disable*) tape compression in DIVA Command.

When tape compression is enabled, any empty tape assigned to the group will have compression enabled, and instances written to the tape will be compressed. Tapes assigned to the group before compression was enabled remain uncompressed, and instances written to the uncompressed tape will be uncompressed.

When exporting a tape, compression is tracked using the isCompressionEnabled attribute. This attribute value can be either true or false.

When writing to a group with a compression setting (*enabled or disabled*), only used tapes with the same compression setting, or empty tapes associated with the set linked to the group, will be used.

Tape Group Encryption

Starting with the DIVA Core 8.0 release, tape drive encryption securely supports bulk tape migration between DIVA Core systems.

See the Installation and Configuration Guide for detailed configuration information.

Sony ODA Drives

DIVA Core supports the Sony optical drives and their WORM media (*using a UDS format*). You can only write AXF formatted objects to Blu-ray discs. The drives are controlled by the Robot Manager and the media is viewed as a tape cartridge.

These drives are shown as ***Unknown Medium Changer*** under the **Medium Changer** devices section in the Windows Device Manager 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.

Using Optical Drives and Discs

The following list is additional information related to the use of the optical drives and discs:

- In the Control GUI the optical discs are displayed under the **Drives** tab.
- Write-Once media must be finalized, and therefore zero remaining space will be reported to the Manager.
- Objects are spanned when there is 100 MB remaining on the disc so that there is enough space left to finalize the disc. After an object is spanned, the disc is considered full and is automatically finalized.
- The Datahub will auto-finalize the discs when there is 500 MB of space remaining (*unless an object was spanned*). However you can finalize a disc manually through the *Optical Disc Archive Utility*.
- If you manually mount a drive, and view it in Windows Explorer, the numeric value at the beginning of each object's file name identifies the object's location on the tape.

Tape Spanning

When the capacity of a tape group begins to reach full capacity (*that is, the group's associated Set ID has no more blank tapes to draw upon*), DIVA Core may attempt to maximize the storage utilization of the existing tapes in the group by filling the remaining free space of each tape by segmenting the object across two or more tapes (*tape spanning*).

By default, the tape spanning feature is configured in the Manager Configuration (*file*) to not span tapes. If an object cannot be spanned across the remaining free space of two tapes within that group the request will be terminated by DIVA Core. You can configure tape spanning to span across more than two tapes at your site, or disable it altogether in the manager.conf configuration file.

During the restore of a spanned object, DIVA Core mounts all associated spanned tapes and automatically joins the spanned object back together. It cannot do this directly, and must copy all segments of the spanned file to a cache disk first. Therefore, restoring a spanned object must use a **Cache Only** or **Cache and Direct** QOS. A **Direct** QOS results in the request terminating.

For Write-Once media, objects are spanned when there is 100 MB remaining so that there is space left and the disc can be finalized. After an object is spanned the disc is considered full and is automatically finalized.

If spanning is disabled and an object is to large to fit on the selected tape, DIVA Core will retry to fit all content on a single empty tape. You can configure the retry logic. You can optionally retry with a used tape with the least or most remaining space.

Note: Tape spanning is not compatible with the **Associative Copy** function.

Note: Any request that writes to a tape linked to a clone, will terminate on span. In addition, any attempt to clone a tape with spanned instances will abort.

If spanning is disabled and the content is too large to fit on the currently selected tape, DIVA Core will retry to fit all contents on a single tape. The type of tape DIVA Core selects on retry is configurable in the Manager configuration file through the DIVAMANAGER_RETRY_ON_SPAN_REJECTED_ALGORITHM configuration parameter.

Protected Mode

When a tape is ejected from the library it is automatically set to *Protected Mode*. When this attribute is set it prevents further archive operations from being performed on the tape and prevents the tape from being repacked.

DIVA Core assumes that when a previously ejected tape is reinserted into a library to perform a restore operation it will then subsequently be ejected and put back into offline storage. Without the *Protected Mode* feature new objects may get written to the tape while it is temporarily in the library and prevent it from being ejected without first moving these required objects to another tape.

Write operations on a protected tape are not allowed unless the *protected* attribute is set back to false in DIVA Command after the tape is reinserted into the library. This attribute does not prevent delete operations on instances located on these tapes (*whether internalized or externalized*).

You may also need to reset this attribute on a tape if the tape was mistakenly ejected from a library, or if the tape was stuck in a tape drive and removed by opening the library door and manually ejected from the library. When the library is then resynchronized with the DIVA Core database the missing tape will be deemed externalized and *Protected Mode* set to true (*the tape is protected*).

Tape Label Management

When a tape is first mounted and objects written to it, DIVA Core writes a *Label* to the beginning of that tape. The label contains important information relating to the management of objects written to, or deleted from, the tape during archive operations. From an operational perspective the most important information in the tape label is the *Barcode Number* of that tape. The barcode is an alphanumeric number on the physical label adhered to the back of the cartridge, and is also written to the label on the magnetic media of the tape.

Whenever a tape is mounted DIVA Core automatically checks the label written on tape to verify that it matches the tape barcode label it instructed the tape library to mount.

This mechanism provides the following two safety features:

- Confirmation that the mapping between the physical drives in the library matches that of the logical connections to each tape drive from the Datahub. This prevents data being written to the wrong tape if there is a configuration mismatch between the physical drives in the library.
- Prevents tapes with foreign labels (*that is, tapes previously used by another archive system*) from being overwritten in error. This behavior is for environments where DIVA Core shares a library with another archive application and the tapes used by that archive application have not been set to Set ID 99.

If DIVA Core identifies a mismatch between the expected label and that of the tape, it will generate an I/O Label Error, and the tape will be set to *Not Writable* and will not be selected for any further write operations.

Media Storage Formats

This section discusses the media formats available in DIVA Core.

AXF Disk and Tape Storage Formats

AXF (*Archive eXchange Format*) is an open format that supports interoperability among disparate content storage systems and ensures the content's long-term availability regardless of 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 your valuable 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.

Native File and Folder Support

Users can see their files and folders in native format on archive devices rather than as an AXF container files. You can also access files and folders on storage devices like object storage. This access opens the archive to the use of third party software to perform operations on the archive (*for example, metadata collection, face recognition, transcoding, and so on*).

The following list identifies the different AXF formats available:

AXF_RF_1.1
This format uses the AXF 1.1 structures, but AXF files won’t contain any overhead. This format is supported on disk arrays and object storage only (*not on tape*).

Note: Telestream does not recommend using the AXF_RF_1.1 format with complex objects.

AXF_1.1
This format is compliant with AXF 1.1 standards.

AXF_1.0
This format is compliant with AXF 1.0 standards.

AXF
This is redirected to AXF_1.1.

LEGACY
This is the formal archive format used by DIVA (*index.txt, 00000001, 00000002, and so on*)

Storage Media Format

In DIVA Core, a tape group or disk array has a *Media Format* parameter that indicates which storage media format to use when creating new archived objects. The *Media Format* can be set to either DIVA Core **Legacy** format or one of the **AXF** formats. You can change this setting at any time and it does not influence content already stored. Therefore, it is possible to have more than one storage media format within tape groups and disk arrays.

DIVA Core writes an object instance in one and only 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. In DIVA Core 8.1, an object can contain multiple instances, each of which can be stored in either Legacy or one of the AXF formats.

Complex objects must be stored in AXF format. Because all complex objects are written in the AXF format, every instance of a complex object will be in the AXF format.

Tape Storage Media Format

Although a tape group can contain more than one storage format, an individual tape has (*at most*) one storage media format. DIVA Core assigns the tape media format to an empty tape when it writes the first object to that tape. The tape is assigned the format of the tape group that appears in the request. After the media format for a tape is assigned, it cannot be changed unless all objects on the tape are deleted. After deletion of all objects from a tape, the tape's format becomes unassigned until DIVA Core writes content to the tape again. 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. 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.

In the current release of DIVA Core, 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.

Disk Storage Media Format

Unlike tapes, disks do not have a format. DIVA Core 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.

Object Instances Media Format

Every tape and disk object instance is assigned a format of Legacy or AXF. The format of a tape or disk instance is assigned when the instance is created and is the format of the tape on which the instance resides. All instances on a tape must have the same 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.

Groups or arrays used by complex object requests must be in an AXF format because complex objects cannot be stored in Legacy format. Therefore, any instance of a complex object will be in the AXF format.

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.

Objects

Each asset that is archived to DIVA Core is called an **Object**.

An object is a DIVA Core logical container for all files consisting of an asset from the original source. Assets from some sources may have separate video, audio and metadata files. When archived in DIVA Core all of these files are referenced as a single object. When the object is restored to a destination, all files that were originally associated with that asset are automatically restored by DIVA Core.

An object is uniquely identified in DIVA Core by its name and category. The object name does not necessarily need to match that of the source file being archived. DIVA Core will always restore the archived files as they were archived, regardless of the DIVA Core object name. Therefore, the same source file can be archived more than once in the same category, if each instance has a unique object name.

Note: Object Names cannot begin with a dollar sign ($).

After an object exists within DIVA Core, it cannot be replaced unless it is first deleted. If an Archive request uses the same name and category of an existing object, DIVA Core will automatically stop the request. However, multiple copies (*or instances*) of an object can be created after the asset is archived.

If a source asset is to be stored in a variety of encoding formats (*for example, MPEG2 Long GOP, DV50, or low resolution proxies*), you can use specific categories to archive the same object based on its encoding format.

Complex Objects

When the Metadata Database feature is enabled, the *complex object* feature is available. DIVA Core can track significantly more than the 10,000 file per object limit set for non-complex objects using complex objects. The actual amount scales with system processing power and storage capacity. A complex object stores more information about the files and folders in an archive, such as subtotals for each directory.

When an object is archived, DIVA Core determines whether the new object should be complex or non-complex based on its number of components (*files*). If the number of components is greater than 5,000 (*the default complex object threshold - configurable*), the object becomes a complex object: otherwise, the object is non-complex. When an object is deemed 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.

Complex versus Non-Complex Objects

A **Complex Object** differs from a non-complex object in some key ways. For example, the file and folder metadata information of a complex object is stored in the Metadata Database (MDS) not in the Oracle database. The file contains the file names, folder names, checksums, and files sizes. The directory that contains these files is the *Metadata Database Root Directory* (*the following section describes how to configure this parameter*). A complex object must be stored in AXF format either on tape or on disk.

A complex object can contain hundreds of thousands of files. In the Control GUI the entire set of files on a tape are not displayed in the Object Properties and Tapes dialog boxes - only a single *placeholder* file is shown to represent the complex object.

Not all DIVA Core operations are supported for complex objects. For example, the **Delete on Source** feature is disabled for complex objects. The checksum features **Verify on Archive** and **Verify on Restore** are also disabled for complex objects. DIVA Connect does not currently support replication of complex objects.

Certain API operations used in Avid Connectivity (*such as* GetByFilename *and* DeleteByFilename) are not currently supported for complex objects.

A complex object maintains information about the folders and files in the archive. Complex objects store subtotals for each folder, including the total number of files and subdirectories within the folder, and the total size of all files within the folder and any subfolders.

The *Complex Object Threshold* is a configurable parameter used by DIVA Core to determine whether a new object should be complex. If a new object has many components (*files*) that exceeds the threshold, the object automatically becomes a complex object. This value is set in the manager.conf configuration file. *Telestream recommends leaving the threshold at the default value (5000 components) unless there is a specific reason to adjust the value*.

Metadata Database (*MDS*)

To effectively operate with large volumes of files and folders and other metadata, DIVA Core stores the metadata separately from the Oracle database in the *DIVA Core Metadata Database*. The DIVA Core metadata database contains files stored in a file system local to the Manager. The directory that contains these files is the *Metadata Database Root Folder*.

The metadata database has very high performance and almost unlimited scalability. The Metadata Database should be treated with the same caution as the Oracle database, and should be backed up at regular intervals using the DIVA Core Backup Service. The Metadata Database is backed by MongoDB.

Complex Objects and FTP

When archiving Complex Objects with the FTP protocol and using FileZilla with default settings, the transfer will typically fail when archiving any object with more than approximately 3900 files. There are two reasons for this possible failure:

- The Datahub connection times out before the size of the object can be computed.
- A request stops in the middle of the transfer because the FTP server (*for example a FileZilla server*) is consuming all of the available sockets.

Note: Telestream only supports Linux-based FTP servers on DIVA Core systems running in the Linux environment, not FileZilla and IIS FTP servers.

Datahub connection timeouts can be resolved by setting the following two parameters either in the ***Server Command Options***, or in the options of the command itself as follows:

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

Telestream also recommends setting the corresponding parameters in the FileZilla server under the ***General Settings***:

```
Connections Timeout = 600
No Transfer Timeout = 1200 (this is the default)
```

If a termination occurs, which may happen during transfer, there are two registry parameters that must be created or modified (*typically created*):

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

Contact Telestream Support for more information on these parameters and to make FTP server and computer registry changes if no qualified personnel are on site.

DIVA Connect Complex Object WAN Transfers

DIVA Core has (*optional*) WAN acceleration functionality built in that allows it to take full advantage of long distance, high latency, network paths (*such as private site to site links or the public internet*), and can perform transfers of complex objects efficiently using the *Data Expedition MTP/IP* protocol.

Example:

The procedure is as follows:

1. DIVA1 restores the complex object to the DIVA2 system by first creating a new AXF file on the DIVA2 system's Data Expedition server.
2. DIVA1 restores all of the files from the local storage to the new AXF file created on the DIVA2 Data Expedition server.
3. The DIVA2 system creates a new AXF file on the destination (*tape, disk, and so on*).
4. The DIVA2 system archives all of the files from the Data Expedition AXF file (*created by* DIVA1 *on the Data Expedition server*) to the newly created AXF file on the destination.

See the DIVA Connect Installation, Configuration, and Operations Guide or contact Telestream Support for more information and assistance (if necessary).

Object Instances

The storage managed by DIVA Core falls into three distinct categories:

- Online Storage (*tapes within a library*)
- Nearline Storage (*disks and cloud storage*)
- Offline Storage (*externalized tapes*)

The name and category of an object in DIVA Core must be unique. However, multiple copies of that object can be created in one or all three of the above classes. Each copy of an object (*including the original archived object itself*) is known as an **Object Instance**.

Apart from creating backup copies, the object instances concept also allows lifecycling of material within DIVA Core. An object may initially be created in online storage for rapid access and also backup instances created on one or more tapes. When the object is no longer required for online or Nearline access the disk instance can be deleted and the tape externalized. Automatic lifecycling of objects, based on their age and location within the archive, can be provided by the SPM (*Storage Plan Manager*) option.

The first instance of an object is created when it is first archived to DIVA Core. Additional instances of the archived object can then be created with the **Copy** and **Associative Copy** commands.

An additional instance of an object cannot be created by re-archiving the original object with the same name and category. This request will be automatically stopped by DIVA Core with the *Object already exists within DIVA Core* error.

Instances are initially numbered sequentially with the original object that is archived to DIVA Core being *Instance 0*. As new instances are created and older instances deleted, the instance numbering may no longer be sequential when an object's properties are viewed in the Control GUI's ***Objects View*** (*under the **Manage** tab*). However, an instance number from a previously deleted instance may be subsequently reused by DIVA Core in additional copy requests.

The following restrictions apply to creating new instances of an object within DIVA Core:

- A tape group can contain two instances of the same object if both can be located on separate tapes. If no additional tapes for that group are available to store the second instance the copy request is terminated.
- A disk array can contain two instances of the same object if both can be located on separate disks within the array. If no additional disk is available, the copy request is terminated.

When an object has multiple instances within the archive and a Restore request is issued, DIVA Core will perform as follows:

- If no instance number is specified in the request, DIVA Core will choose the instance that will allow the request to complete in the shortest possible time. A disk instance is preferred over a tape instance. However, a tape instance may be selected in some configurations if the QOS specified in the request is **Cache Only** or **Cache and Direct**.
- If no instance number is specified in the Restore request and a disk instance exists, but the disk is offline, the tape instance will automatically be selected.
- If two or more instances are present on tape and no disk instances exist, and one tape is currently in use by another request (*or is externalized*), the tape containing the other instance will automatically be selected.
- If two or more instances exist on tape, and a read error occurs on the first instance selected, the request will automatically be attempted on the other instances until the request can be completed successfully. If no instances can be read, the request will be terminated.
- If a specific *Instance Number* is specified in the restore request, DIVA Core will only use that instance. If the media containing the instance is offline (*for disks*), externalized (*for tapes*), or an I/O or read error occurs, the request will be terminated.

Requiring and Releasing Instances

Requiring and releasing instances enables an application, such as a third party MAM (*Media Asset Management*) system, or a DIVA Core user, to flag the DIVA Core objects (*or instances*) that are externalized but need to be restored (***Required***), and which instances are no longer needed and can be externalized (***Released***). The release mechanism is a more precise alternative to a group externalization approach for externalizing instances.

The ***Required Release*** view in the Control GUI's **Manage** tab is provided to check instances whose internalized/externalized status is in discrepancy with their **Released/Required** status. This view also provides a fast easy method for identifying which tapes are to be entered in the library or can be externalized.

By default, object instances are assumed to be available in DIVA Core. The **Release** command must be invoked on instances before ejecting their corresponding tapes. However, the **Eject** command provides an option that automatically performs the release on every instance entirely located on the tapes to be ejected.

After being created by a Copy or an Archive request, an instance is assumed to be required to be available, so its status is INSERTED and REQUIRED. Executing a **Require** command on a released instance results in a *Required Instance*. Correspondingly, releasing a required instance results in a *Released Instance*.

Requests

A request is a command that is issued to DIVA Core to perform an operation. Requests can be issued through the Control GUI or an Archive Initiator.

The most common request types are for transferring content to the archive (*referred to as an Archive request*), or transferring content out of the archive (*referred to as a Restore or Partial File Restore request*).

You use other request types for managing the objects within the archive once they have been created. Examples of other request types are Copy, Delete, and Repack Tape requests.

Each request is automatically given a unique identifier by DIVA Core (*called the Request ID*), which you can use later to retrieve event logs or other properties of each request. DIVA Core stores the records of up to 50,000 requests in its database.

Because multiple requests may be received simultaneously by DIVA Core, they are all placed into a queue and are executed on a first come, first served basis. The execution order of requests can be prioritized using the *Request Priority* parameter. The **Current Requests** frame of the **Manager View** in the Control GUI displays the queue of requests that are currently being processed by DIVA Core.

When restoring the same file to the same destination twice in parallel, the behavior on Windows and Linux is different. On Windows, the first restore (*they cannot arrive exactly at the same time*) will lock the file so that the second one will terminate. On Linux, there is no such lock at the file system level. Both restores are executed at the same time, and both will write to the same file. The content of the resultant file is not predictable.

The DIVA Core available ***Request Options*** are as follows:

Archive Requests

-delete_on_source

Restore Requests

-do_not_overwrite

-do_not_check_existence

-delete_and_write

Request Options take precedence over the normal ***Additional Service*** specification. Also, the normal ***Additional Service*** specification takes precedence over the ***Server Connect Options***.

You can also specify the ***Additional Services*** available for a Restore request in the ***Server Connect Options***. If specified, the Server will use the ***Additional Service*** setting as a default. This can be overridden by specifying the ***Additional Service*** at a request level in the normal way, or as a new ***Request Option***. Because these connect options are specific to a Restore request, the options are ignored for any other type of requests using the Server.

Request Types

This section describes the various DIVA Core request types available.

When connected to the Manager, you access the **Action** tab from any view in the Control GUI. This tab enables you to execute requests to be issued to DIVA Core. You can use a third party initiator application (*for example, an Automation System*) instead of, or in addition to, the Control GUI interface. The options in this area are only accessible when you are logged in with the *Administrator* profile.

The different requests available under the Control GUI's **Action** tab are as follows:

Archive
Copies a file from a Source Server to DIVA Core.

Delete
Deletes all instances, or a selected instance, of an object.

Require
Sets an object's status to **Required**. The associated tape must be inserted into a managed library.

Release
Sets an object's status to **Released**. After an object is released, it can be externalized.

Cancel
Cancels a previously submitted request by either specifying the ***Request ID*** or selecting a request beforehand in the *Current Requests* view.

Change Priority
Increases or decreases the pending requests scheduler priority.

Assign Storage Plan
Assigns a Storage Plan to the selected object.

Restore
Copies a file from DIVA Core to a single destination.

Partial Restore
Copies only part of a file (*based on timecode, byte offsets, folders, or DPX frames*) from DIVA Core to a Destination Server.

Multiple Restore (or N-Restore)
Restores an object from DIVA Core to more than one destination simultaneously.

Copy
Enables an existing object to be copied to another group.

Copy As
Enables an existing object to be copied to another name, group, or category.

Associative Copy
Enables multiple objects from various locations in the archive to all be copied to a single tape with a single command.

Repack Tape
Issues a manual Repack request for the selected tape.

Verify Tape
Issues a Verify Tape request for the selected tape.

Insert Tape
Used to insert tapes into a DIVA Core library through its **CAP (Cartridge Access Port)**.

Eject Tape

Ejects the selected tape(s) from the library to the CAP.

Export Tape

Enable exporting a tape (*and its objects*) from one DIVA Core system to another.

Migrate Content

Transfers the existing content of a tape group to another group or array.

Automatic Repack

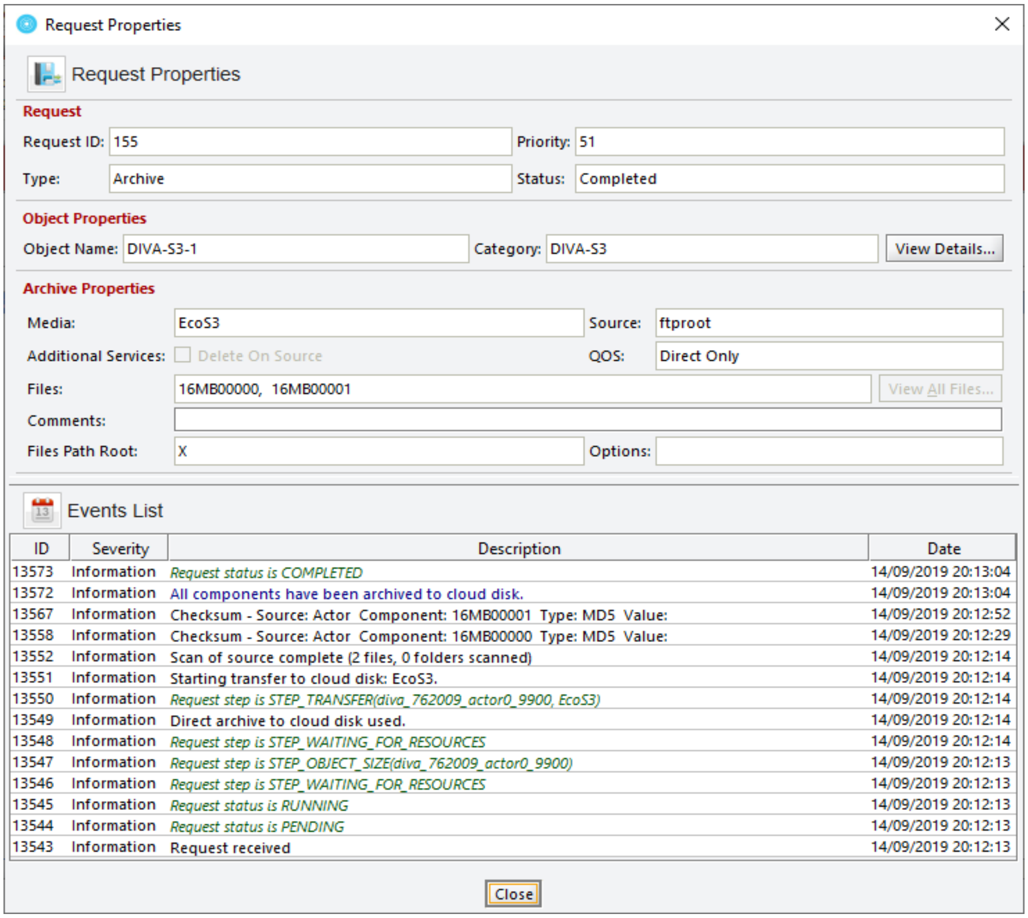
Issues an Automatic Repack request for the selected tape.

Amazon S3 Transfers

The following subsections describe transfers to and from Amazon S3 disks.

Archiving from a Non-S3 Source to an S3 Disk

DIVA Core allows a user to archive content from a regular source such as an FTP server to an Amazon S3 disk. The following figure displays the *Request Properties* screen for this type of transfer:



Request

Request ID: 155
Priority: 51
Type: Archive
Status: Completed

Object Properties

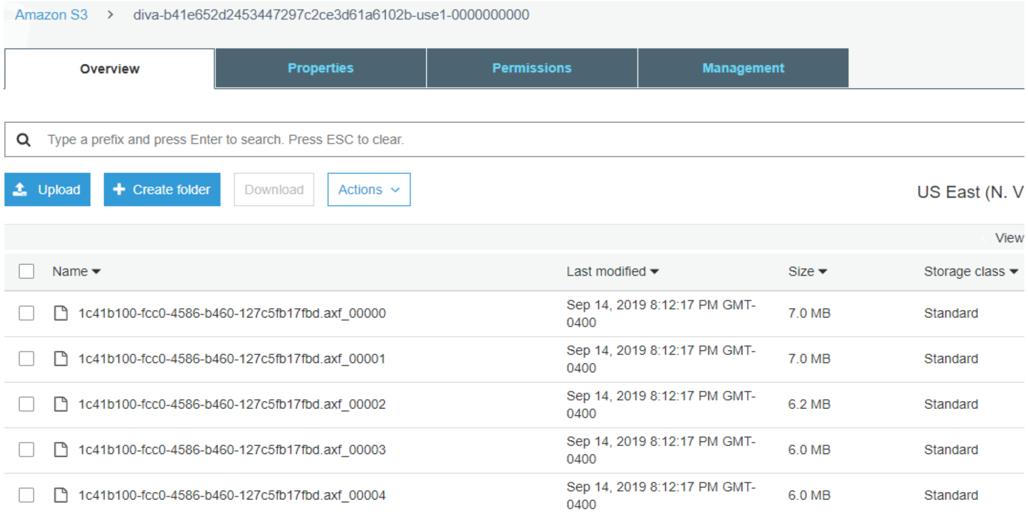
Object Name: DIVA-S3-1
Category: DIVA-S3

Archive Properties

Media: EcoS3
Source: ftproot
Additional Services: □ Delete On Source
QOS: Direct Only
Files: 16MB00000, 16MB00001
Comments:
Files Path Root: X
Options:

ID	Severity	Description	Date
13573	Information	*Request status is COMPLETED*	14/09/2019 20:13:04
13572	Information	All components have been archived to cloud disk.	14/09/2019 20:13:04
13567	Information	Checksum - Source: Actor Component: 16MB00001 Type: MD5 Value:	14/09/2019 20:12:52
13558	Information	Checksum - Source: Actor Component: 16MB00000 Type: MD5 Value:	14/09/2019 20:12:29
13552	Information	Scan of source complete (2 files, 0 folders scanned)	14/09/2019 20:12:14
13551	Information	Starting transfer to cloud disk: EcoS3.	14/09/2019 20:12:14
13550	Information	*Request step is STEP_TRANSFER(diva_762009_actor0_9900, EcoS3)*	14/09/2019 20:12:14
13549	Information	Direct archive to cloud disk used.	14/09/2019 20:12:14
13548	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	14/09/2019 20:12:14
13547	Information	*Request step is STEP_OBJECT_SIZE(diva_762009_actor0_9900)*	14/09/2019 20:12:13
13546	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	14/09/2019 20:12:13
13545	Information	*Request status is RUNNING*	14/09/2019 20:12:13
13544	Information	*Request status is PENDING*	14/09/2019 20:12:13
13543	Information	Request received	14/09/2019 20:12:13

Files in a folder of an FTP server are split into 5 AXF segments (*determined by number of Threads Per Transfer specified in the Storage Accounts configuration*) and transferred to an Amazon S3 bucket using 5 threads within the Datahub. DIVA Core automatically creates the bucket named diva-<unique bucket id>-<region>-<index> where the index increases every 100,000 instances. This is specified in the ***Max Instances Per Bucket*** setting of the array's configuration. The unique bucket id was generated on creation of the storage account.

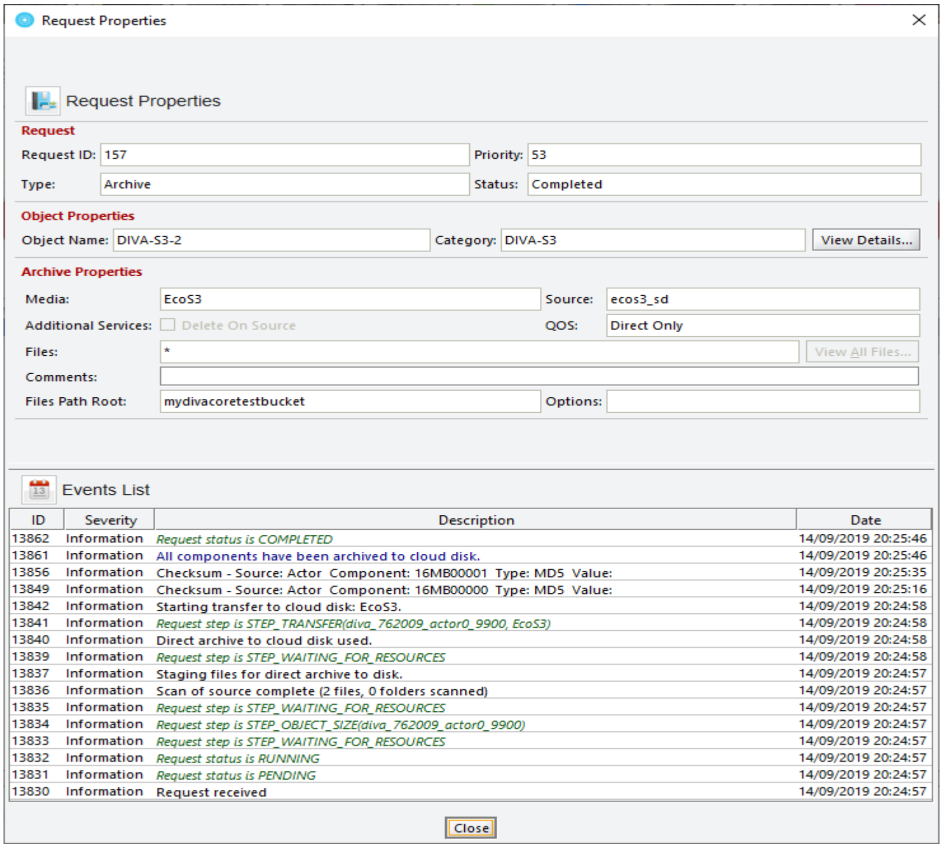


Amazon S3 > diva-b41e652d2453447297c2ce3d61a6102b-use1-0000000000

	Name	Last modified	Size	Storage class
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00000	Sep 14, 2019 8:12:17 PM GMT-0400	7.0 MB	Standard
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00001	Sep 14, 2019 8:12:17 PM GMT-0400	7.0 MB	Standard
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00002	Sep 14, 2019 8:12:17 PM GMT-0400	6.2 MB	Standard
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00003	Sep 14, 2019 8:12:17 PM GMT-0400	6.0 MB	Standard
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00004	Sep 14, 2019 8:12:17 PM GMT-0400	6.0 MB	Standard

Archiving from an S3 Source to an S3 Disk

Users can archive content from an S3 source to an Amazon S3 disk. The following figure displays the *Request Properties* screen for this type of transfer:



Request

Request ID:	157	Priority:	53
Type:	Archive	Status:	Completed

Object Properties

Object Name: DIVA-S3-2 Category: DIVA-S3 View Details...

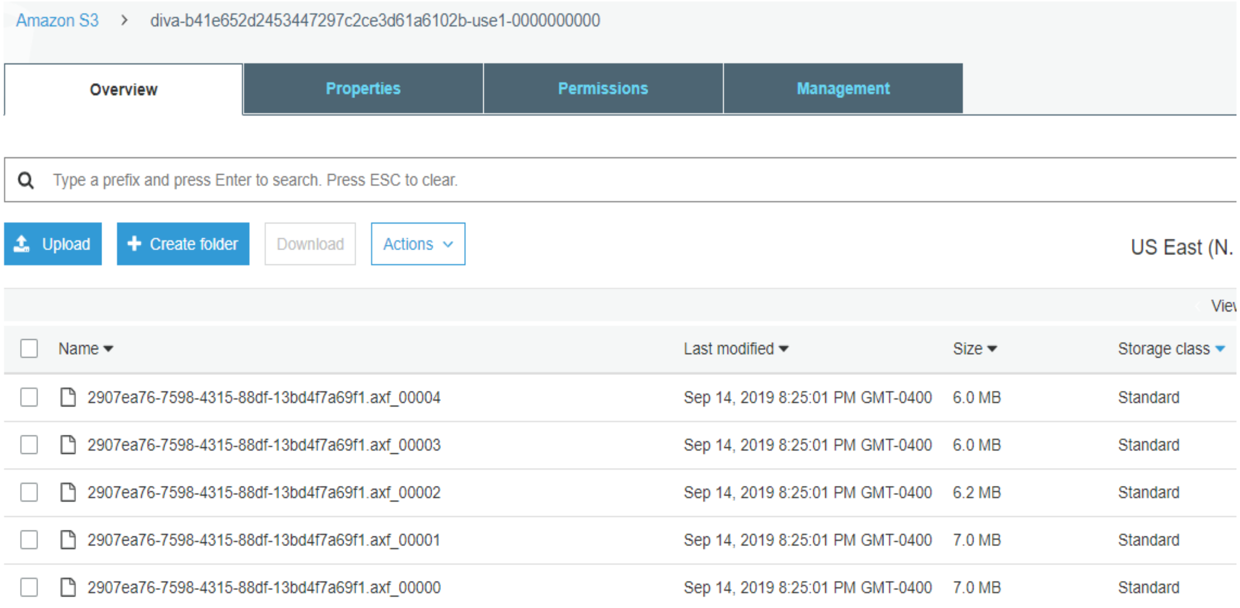
Archive Properties

Media:	EcoS3	Source:	ecos3_sd
Additional Services:	□ Delete On Source	QOS:	Direct Only
Files:	*		View All Files...
Comments:			
Files Path Root:	mydivacoretestbucket	Options:	

Events List

ID	Severity	Description	Date
13862	Information	Request status is COMPLETED	14/09/2019 20:25:46
13861	Information	All components have been archived to cloud disk.	14/09/2019 20:25:46
13856	Information	Checksum - Source: Actor Component: 16MB00001 Type: MD5 Value:	14/09/2019 20:25:35
13849	Information	Checksum - Source: Actor Component: 16MB00000 Type: MD5 Value:	14/09/2019 20:25:16
13842	Information	Starting transfer to cloud disk: EcoS3.	14/09/2019 20:24:58
13841	Information	Request step is STEP_TRANSFER(diva_762009_actor0_9900, EcoS3)	14/09/2019 20:24:58
13840	Information	Direct archive to cloud disk used.	14/09/2019 20:24:58
13839	Information	Request step is STEP_WAITING_FOR_RESOURCES	14/09/2019 20:24:58
13837	Information	Staging files for direct archive to disk.	14/09/2019 20:24:57
13836	Information	Scan of source complete (2 files, 0 folders scanned)	14/09/2019 20:24:57
13835	Information	Request step is STEP_WAITING_FOR_RESOURCES	14/09/2019 20:24:57
13834	Information	Request step is STEP_OBJECT_SIZE(diva_762009_actor0_9900)	14/09/2019 20:24:57
13833	Information	Request step is STEP_WAITING_FOR_RESOURCES	14/09/2019 20:24:57
13832	Information	Request status is RUNNING	14/09/2019 20:24:57
13831	Information	Request status is PENDING	14/09/2019 20:24:57
13830	Information	Request received	14/09/2019 20:24:57

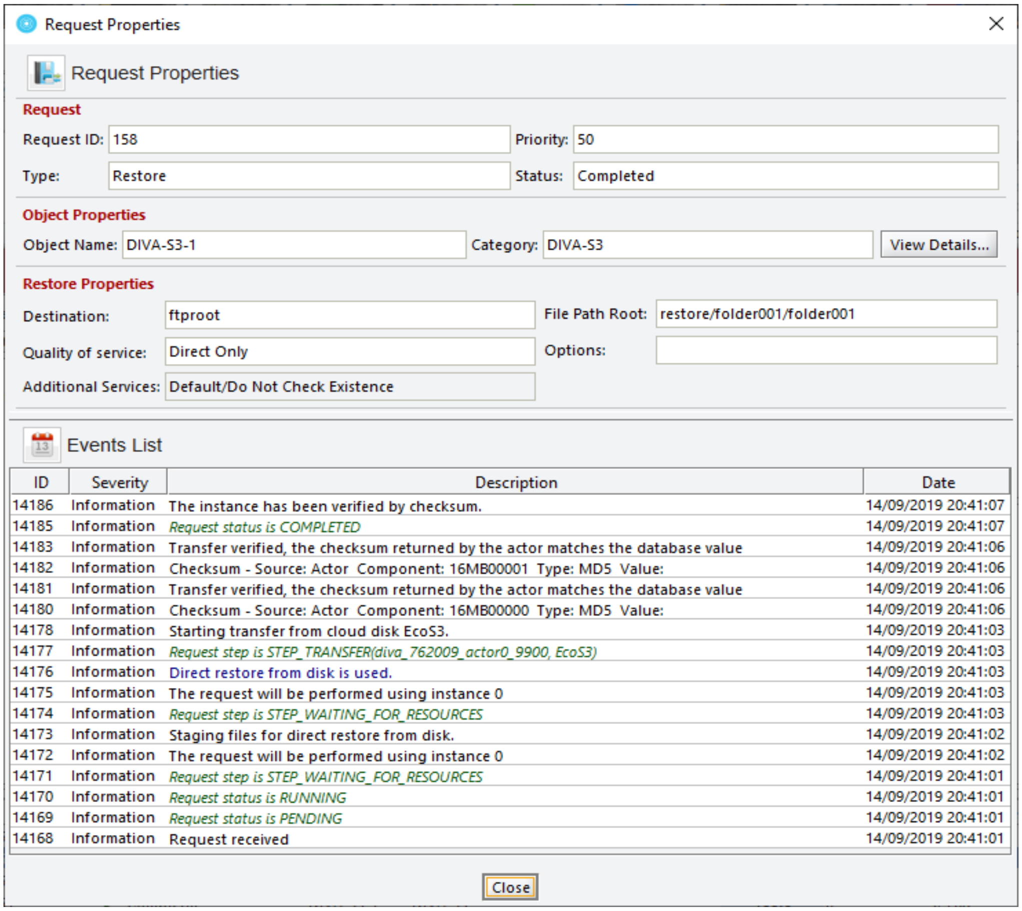
Again the files are split into AXF segments as shown in the following figure:


Amazon S3 > diva-b41e652d2453447297c2ce3d61a6102b-use1-0000000000
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
US East (N.
Viev

□	Name	Last modified	Size	Storage class
□	2907ea76-7598-4315-88df-13bd4f7a69f1.axf_00004	Sep 14, 2019 8:25:01 PM GMT-0400	6.0 MB	Standard
□	2907ea76-7598-4315-88df-13bd4f7a69f1.axf_00003	Sep 14, 2019 8:25:01 PM GMT-0400	6.0 MB	Standard
□	2907ea76-7598-4315-88df-13bd4f7a69f1.axf_00002	Sep 14, 2019 8:25:01 PM GMT-0400	6.2 MB	Standard
□	2907ea76-7598-4315-88df-13bd4f7a69f1.axf_00001	Sep 14, 2019 8:25:01 PM GMT-0400	7.0 MB	Standard
□	2907ea76-7598-4315-88df-13bd4f7a69f1.axf_00000	Sep 14, 2019 8:25:01 PM GMT-0400	7.0 MB	Standard

Restoring from an S3 Disk to a Non-S3 Destination

Users can restore an object on an Amazon S3 disk to a non-S3 destination, for example an FTP server. The following figure displays the Request Properties screen for this type of transfer:



Request

Request ID:	158	Priority:	50
Type:	Restore	Status:	Completed

Object Properties

Object Name:	DIVA-S3-1	Category:	DIVA-S3

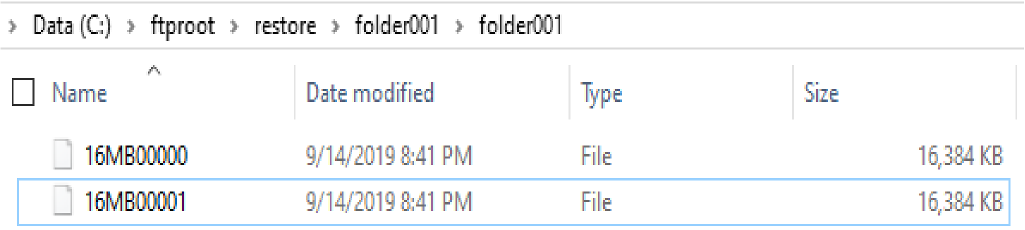
Restore Properties

Destination:	ftproot	File Path Root:	restore/folder001/folder001
Quality of service:	Direct Only	Options:	
Additional Services:	Default/Do Not Check Existence		

Events List

ID	Severity	Description	Date
14186	Information	The instance has been verified by checksum.	14/09/2019 20:41:07
14185	Information	*Request status is COMPLETED*	14/09/2019 20:41:07
14183	Information	Transfer verified, the checksum returned by the actor matches the database value	14/09/2019 20:41:06
14182	Information	Checksum - Source: Actor Component: 16MB00001 Type: MD5 Value:	14/09/2019 20:41:06
14181	Information	Transfer verified, the checksum returned by the actor matches the database value	14/09/2019 20:41:06
14180	Information	Checksum - Source: Actor Component: 16MB00000 Type: MD5 Value:	14/09/2019 20:41:06
14178	Information	Starting transfer from cloud disk EcoS3.	14/09/2019 20:41:03
14177	Information	*Request step is STEP_TRANSFER(diva_762009_actor0_9900, EcoS3)*	14/09/2019 20:41:03
14176	Information	Direct restore from disk is used.	14/09/2019 20:41:03
14175	Information	The request will be performed using instance 0	14/09/2019 20:41:03
14174	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	14/09/2019 20:41:03
14173	Information	Staging files for direct restore from disk.	14/09/2019 20:41:02
14172	Information	The request will be performed using instance 0	14/09/2019 20:41:02
14171	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	14/09/2019 20:41:01
14170	Information	*Request status is RUNNING*	14/09/2019 20:41:01
14169	Information	*Request status is PENDING*	14/09/2019 20:41:01
14168	Information	Request received	14/09/2019 20:41:01

The following figure shows the files restored to the FTP server:

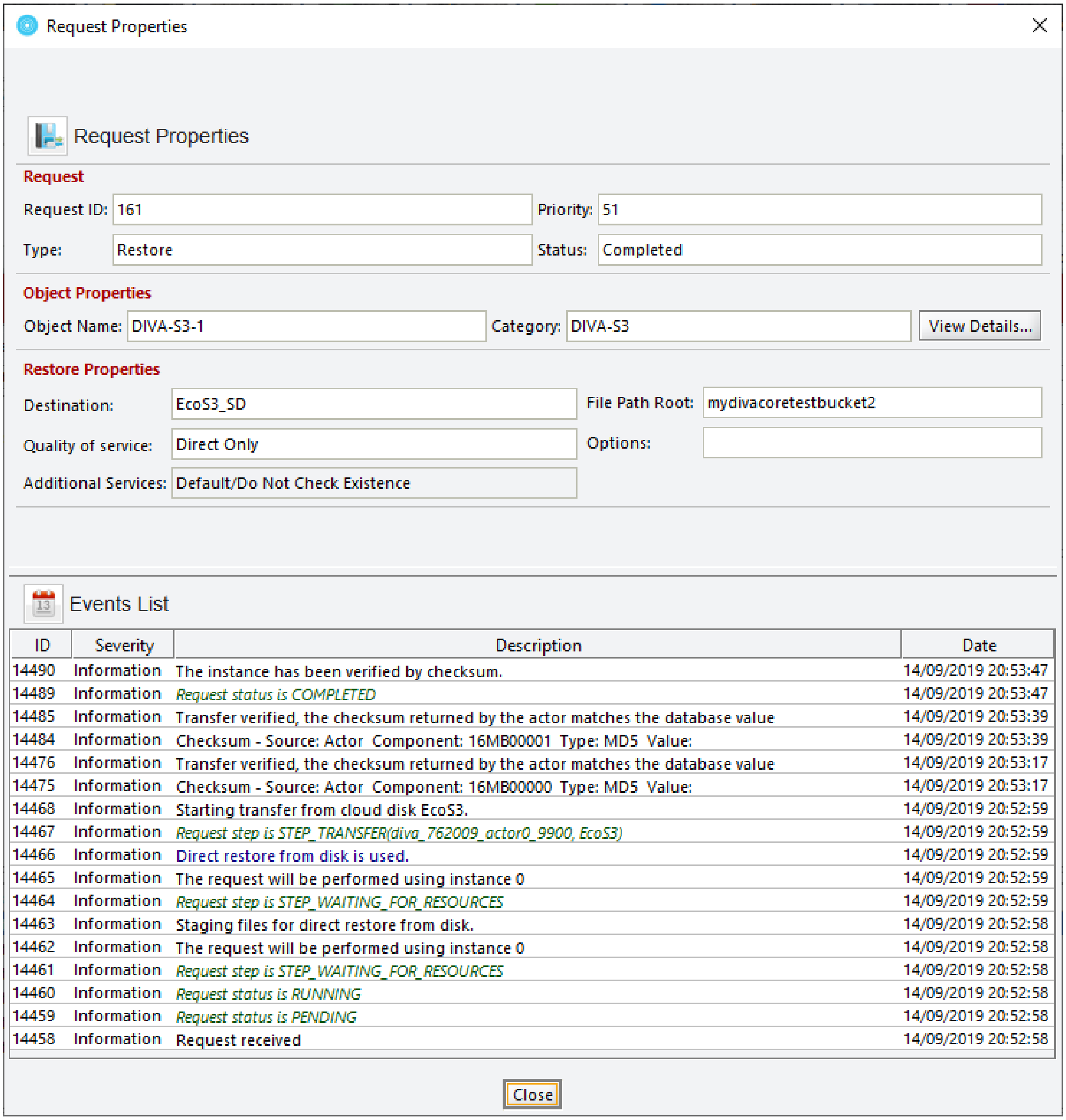


› Data (C:) › ftproot › restore › folder001 › folder001

Name	Date modified	Type	Size
16MB00000	9/14/2019 8:41 PM	File	16,384 KB
16MB00001	9/14/2019 8:41 PM	File	16,384 KB

Restore from an S3 Disk to an S3 Destination

Users can restore an object on an Amazon S3 disk as an AXF file to an S3 destination. The following figure displays the Request Properties screen for this type of transfer:



Request Properties

Request Properties

Request

Request ID: 161
Priority: 51
Type: Restore
Status: Completed

Object Properties

Object Name: DIVA-S3-1
Category: DIVA-S3

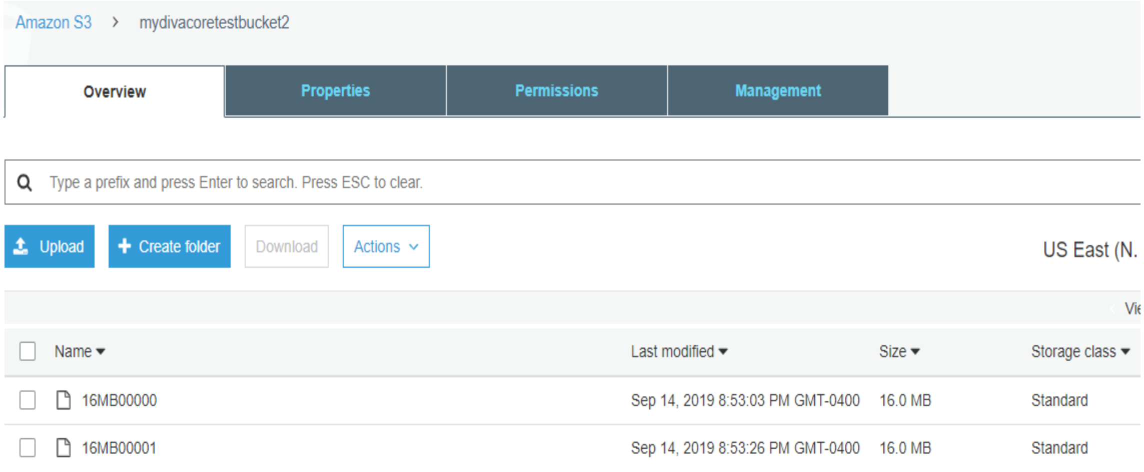
Restore Properties

Destination: EcoS3_SD
File Path Root: mydivacoretestbucket2
Quality of service: Direct Only
Options:
Additional Services: Default/Do Not Check Existence

Events List

ID	Severity	Description	Date
14490	Information	The instance has been verified by checksum.	14/09/2019 20:53:47
14489	Information	*Request status is COMPLETED*	14/09/2019 20:53:47
14485	Information	Transfer verified, the checksum returned by the actor matches the database value	14/09/2019 20:53:39
14484	Information	Checksum - Source: Actor Component: 16MB00001 Type: MD5 Value:	14/09/2019 20:53:39
14476	Information	Transfer verified, the checksum returned by the actor matches the database value	14/09/2019 20:53:17
14475	Information	Checksum - Source: Actor Component: 16MB00000 Type: MD5 Value:	14/09/2019 20:53:17
14468	Information	Starting transfer from cloud disk EcoS3.	14/09/2019 20:52:59
14467	Information	*Request step is STEP_TRANSFER(diva_762009_actor0_9900, EcoS3)*	14/09/2019 20:52:59
14466	Information	Direct restore from disk is used.	14/09/2019 20:52:59
14465	Information	The request will be performed using instance 0	14/09/2019 20:52:59
14464	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	14/09/2019 20:52:59
14463	Information	Staging files for direct restore from disk.	14/09/2019 20:52:58
14462	Information	The request will be performed using instance 0	14/09/2019 20:52:58
14461	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	14/09/2019 20:52:58
14460	Information	*Request status is RUNNING*	14/09/2019 20:52:58
14459	Information	*Request status is PENDING*	14/09/2019 20:52:58
14458	Information	Request received	14/09/2019 20:52:58

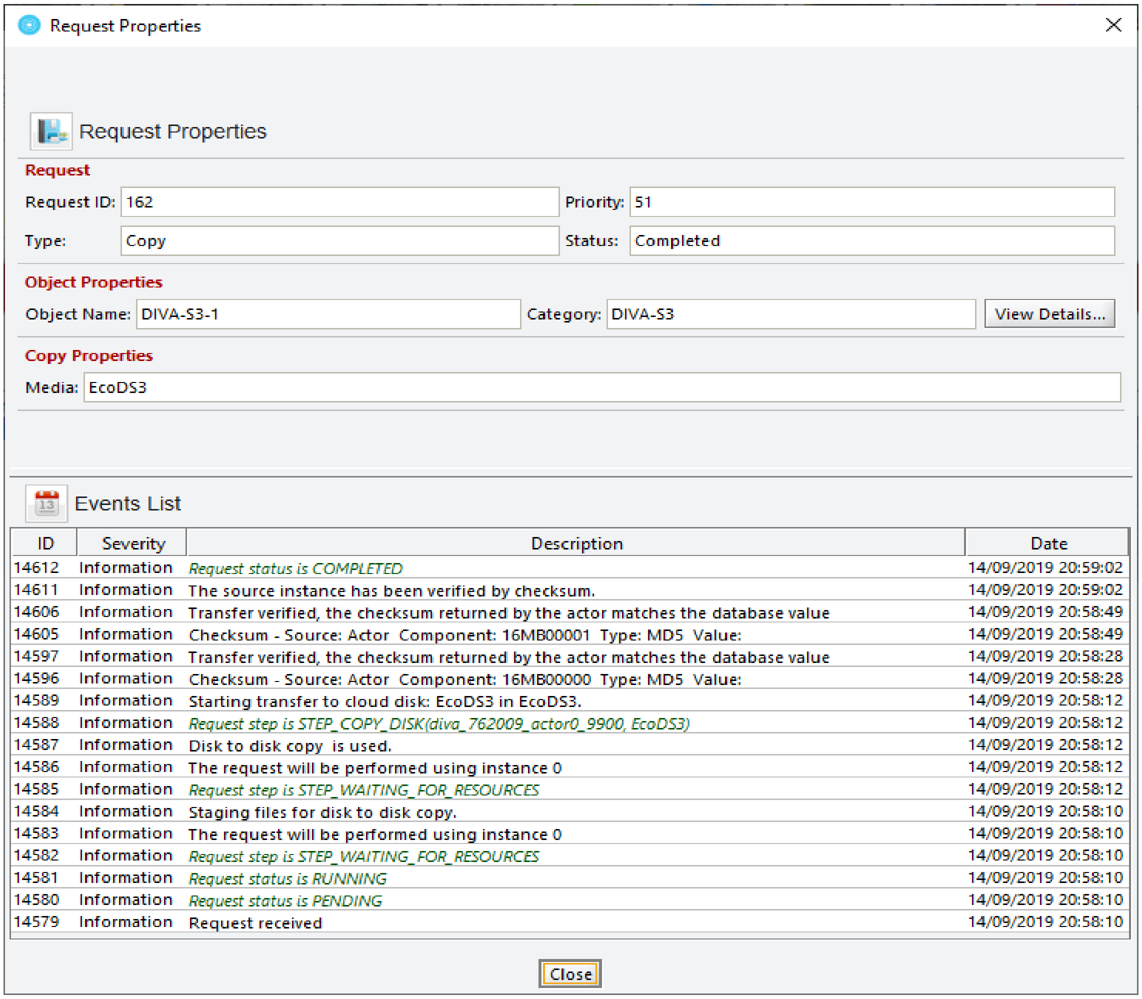
The following figure shows the files restored to the destination:


Amazon S3 › mydivacoretestbucket2
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
US East (N.
Vi

Name	Last modified	Size	Storage class
16MB00000	Sep 14, 2019 8:53:03 PM GMT-0400	16.0 MB	Standard
16MB00001	Sep 14, 2019 8:53:26 PM GMT-0400	16.0 MB	Standard

Copy from an S3 Disk to another S3 Disk

Users can copy an object from one S3 disk to another S3 disk in the same AWS account, or in another AWS account. Here is an example of a cross-account copy. The object previously archived on the EcoS3 account was copied to the EcoDS3 Amazon account.



Request

Request ID: 162 Priority: 51

Type: Copy Status: Completed

Object Properties

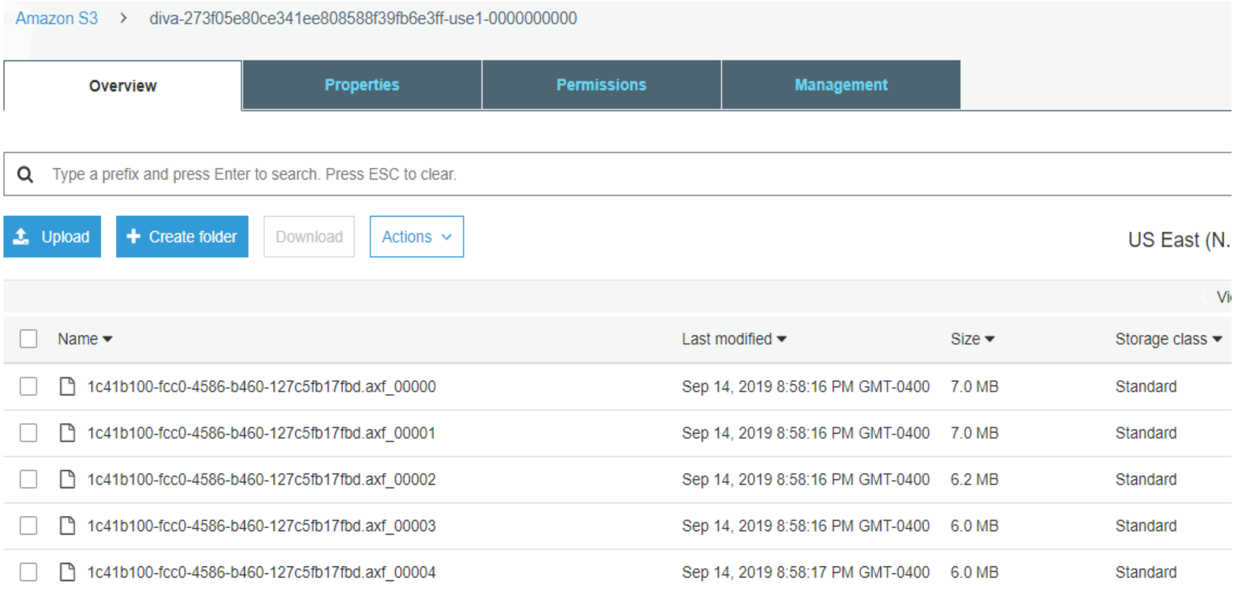
Object Name: DIVA-S3-1 Category: DIVA-S3

Copy Properties

Media: EcoDS3

ID	Severity	Description	Date
14612	Information	*Request status is COMPLETED*	14/09/2019 20:59:02
14611	Information	The source instance has been verified by checksum.	14/09/2019 20:59:02
14606	Information	Transfer verified, the checksum returned by the actor matches the database value	14/09/2019 20:58:49
14605	Information	Checksum - Source: Actor Component: 16MB00001 Type: MD5 Value:	14/09/2019 20:58:49
14597	Information	Transfer verified, the checksum returned by the actor matches the database value	14/09/2019 20:58:28
14596	Information	Checksum - Source: Actor Component: 16MB00000 Type: MD5 Value:	14/09/2019 20:58:28
14589	Information	Starting transfer to cloud disk: EcoDS3 in EcoDS3.	14/09/2019 20:58:12
14588	Information	*Request step is STEP_COPY_DISK(diva_762009_actor0_9900, EcoDS3)*	14/09/2019 20:58:12
14587	Information	Disk to disk copy is used.	14/09/2019 20:58:12
14586	Information	The request will be performed using instance 0	14/09/2019 20:58:12
14585	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	14/09/2019 20:58:12
14584	Information	Staging files for disk to disk copy.	14/09/2019 20:58:10
14583	Information	The request will be performed using instance 0	14/09/2019 20:58:10
14582	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	14/09/2019 20:58:10
14581	Information	*Request status is RUNNING*	14/09/2019 20:58:10
14580	Information	*Request status is PENDING*	14/09/2019 20:58:10
14579	Information	Request received	14/09/2019 20:58:10

The following figure shows the AXF segments created in another bucket of a different AWS account:


Amazon S3 > diva-273f05e80ce341ee808588f39fb6e3ff-use1-0000000000
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
US East (N.
Vi

□	Name	Last modified	Size	Storage class
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00000	Sep 14, 2019 8:58:16 PM GMT-0400	7.0 MB	Standard
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00001	Sep 14, 2019 8:58:16 PM GMT-0400	7.0 MB	Standard
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00002	Sep 14, 2019 8:58:16 PM GMT-0400	6.2 MB	Standard
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00003	Sep 14, 2019 8:58:16 PM GMT-0400	6.0 MB	Standard
□	1c41b100-fcc0-4586-b460-127c5fb17fbd.axf_00004	Sep 14, 2019 8:58:17 PM GMT-0400	6.0 MB	Standard

Note: It is also possible to copy from an OCI Disk in an Oracle Object Storage Account to an Amazon S3 Disk in an Amazon account.

Oracle Storage Cloud Transfers

The Oracle Storage Cloud is an object storage solution that offers two types of accounts usable with DIVA Core: *metered* and *non-metered* accounts. Visit http://docs.oracle.com/cd/E60880_01/VLPFN/whatis.htm#BABDADAE for information on Oracle Storage Cloud storage accounts.

The non-metered account enables the creation of standard class containers. Objects written inside standard containers are accessible immediately at any time.

With a metered account, DIVA Core can archive to standard and also to archive class containers. With archive containers, objects written to a deep archive storage device require a restoration process before they can be downloaded.

An object located in deep archive requires a maximum of 4 hours to restore to a configured Destination Server because the content will first be transferred from tape to Cloud cache, and then transferred from cache to the final destination.

When a Restore request is made for an object with a cloud instance, DIVA Core will *always* attempt to restore a local (*non-cloud*) instance of an object. If all local instances are offline, no local instances exist, or when a cloud instance is explicitly requested (*a Restore Instance request*), then DIVA Core will restore from a cloud instance.

Only Datahubs configured for **CLOUD ARCHIVE** can transfer content to the cloud. Only Datahubs configured for **CLOUD RESTORE** can transfer content from the cloud.

Object Storage Destinations

DIVA Core 8.1 enables restoring content to a destination, and archiving content from a source, linked to an Oracle Object Storage account. You can restore any type of object to these destinations. However, these destinations do not support symbolic links.

The ***Files Path Root*** for the destination must contain a value, and can contain an optional prefix. The value identifies the name for the target container. You use the optional prefix if you do not want to restore to the container root directory. The prefix must be separated from the container name using either / or \. For example, container, container\folder, and container/subdir1/subdir2 are all valid paths.

EMC ECS Object Store Integration

DIVA Core 8.1 supports local arrays that include disks with *Swift* interfaces, for example EMC ECS Object Store.

During an upgrade from an earlier DIVA Core release, all disk instances with an **ARCHIVE** or **STANDARD** Storage Class are updated with a storage option containing -storage_location=CLOUD and -oracle_storage_class={ARCHIVE|STANDARD}. All disk instances with a NONE Storage Class are updated with a storage option containing -storage_location=LOCAL and -oracle_storage_class=NONE. All Datahub-Disk connections with **cloud** as the interface are updated to **Swift** for the interface.

See the Installation and Configuration Guide for detailed configuration information.

True Remaining Tape Size and Last Written Position

For some specific tape drives (*Oracle T10K and IBM LTO*) the Datahub 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 (*instead of relying on the size of the tape type*) to obtain the true total and remaining size on the tape in all tape based operations.

Export and import operations also now include the total tape size.

Archive Requests

An Archive operation is defined as the transfer of files to DIVA Core. The files are then stored as an object. You issue an Archive request by selecting the **Archive** option from the Control GUI **Action** tab. The request submits an object Archive request to the Manager for processing.

The following fields are included in the Send Archive Request screen:

Object Name

The name of the object to be archived.

Note: Object Names cannot begin with a dollar sign ($).

Category

The category of the object to be archived.

Source

The name of the source (*for example, a video server, browsing server, and so on*). This name must be known to the DIVA Core configuration.

Media

This field designates either a group of tapes or an array of disks declared in the configuration where the instance must be created. When this parameter is a null string, the default group of tapes named **DEFAULT** is used.

Files Path Root

The root folder for the files (*see the examples in the following section*).

Storage Plan

This field defines the Storage Plan to use for this object. If no Storage Plan is assigned the default Storage Plan will be used.

Add. Service

Select this check box to delete the original file after it has been archived.

Note: Delete on Source is not supported for Broadcast Servers.

Quality of Service

One of the following codes (*see the* Quality of Service *section in this chapter for detailed descriptions*):

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

DIVA_QOS_CACHE_ONLY
Use cache archive only.

DIVA_QOS_DIRECT_ONLY
Use direct archive only - no disk instance is created.

DIVA_QOS_CACHE_AND_DIRECT
Use cache archive if available, or direct archive if cache archive is not available.

DIVA_QOS_DIRECT_AND_CACHE
Use direct archive if available, or cache archive if direct archive is not available.

Additional and optional services are available. To request those services, use a logical `OR` between the previously documented Quality of Service parameter and the following constant:

DIVA_ARCHIVE_SERVICE_DELETE_ON_SOURCE
Delete source files when the tape migration is done. Available for local sources, disk sources, and standard FTP sources. *This feature is not available for complex objects.*

Priority

The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Files
The file names to be archived from the source. If multiple file names are specified, all are referenced by the object name.

Comments
Optional information describing the object. This field is optional and can be left empty.

Options
Additional options for performing the transfer of data from the Source Server to DIVA Core. These options supersede any options specified in the DIVA Core configuration database. Currently the possible values for *Options* are as follows:

No Entry
No entry in this field specifies no options.

-r
Using **-r** specifies that every name in *filenamesList* that refers to a folder must be scanned recursively. This also applies when *FilesPathRoot* is specified and an asterisk designates the files to be archived. This option can be used when archiving from a local source or from a standard FTP Server.

-login
A user name and password is required to log in to some sources. This option obsoletes the **-gateway** option from earlier releases.

-pass
The password used with **-login**.

Archive Request Files Path Root and Files Parameters

The ***Files Path Root*** and ***Files*** parameters in the Archive Request window determines the main folder location, and the subfolders and files to be archived. Each serves a different purpose, yet both parameters work with each other. Identify a logical business object before filling in these parameters and executing the request.

The ***Files Path Root*** field identifies the path to the main file folder (*the top folder*). For example, c:\DROPFOLDER\Media\Object1\.

What you enter in the ***Files*** field text box identifies the individual files under the main folder (*the identified **Files Path Root***) and any additional subfolders and files. For example, subfolder1\file3.

The ***Files*** field may contain an absolute path. However, this is not recommended because it prevents the object from being restored to a different root folder.

Assuming a ***Files Path Root*** is identified, do not use the full file path in the ***Files*** field. You must only use the folder names and file names that are located under the identified ***Files Path Root*** folder. Alternatively, the ***Files Path Root*** field can be left blank and the full file path and name may be entered into the ***Files*** field.

The following are examples of how these parameters can be utilized:

Correct Examples

The following entries will archive only the specified files located in C:\DROPFOLDER\Media\Object1\ and the subfolder1\file3.

Files Path Root

C:\DROPFOLDER\Media\Object1\

Files

file1
file2
subfolder1\file3

The following entries will archive all folders and files located in C:\DROPFOLDER\Media\Object1\.

Files Path Root

C:\DROPFOLDER\Media\Object1\

Files

*

The following entries are correct but not recommended, because in the future the object cannot be restored to a different location. The system will lose flexibility and compatibility with other storage devices and in some scenarios, transcoding and Partial File Restore capabilities will also be limited. In this example, the ***Files Path Root*** was left blank and the absolute paths are entered in the ***Files*** field.

Files Path Root

Files

C:\DROPFOLDER\Media\Object1\file1
C:\DROPFOLDER\Media\Object1\file2
C:\DROPFOLDER\Media\Object1\subfolder1\file3

Incorrect Example

The following entries will result in an error and the Archive request will not be completed.

Files Path Root

C:\DROPFOLDER\Media\Object1\

Files

C:\DROPFOLDER\Media\Object1\file1
C:\DROPFOLDER\Media\Object1\file2
C:\DROPFOLDER\Media\Object1\subfolder1\file3

Archive Request with Delete on Source

There are instances where you must delete content, and possibly the parent folder, on a server. There are two options available to satisfy all possible scenarios:

-r
Recursive delete

-delete_fpr
Recursive deletion including the parent folder

The two options work either separately or together as indicated in the following workflow examples:

Example 1

Files Path Root

C:\source\root

Files

*

Options

-r

DIVA Core will delete the content of C:\source\root recursively because of these settings.

Example 2

Files Path Root

C:\source\root

Files

*

Options

-r -delete_fpr

DIVA Core will delete the content of C:\source\root recursively and the parent folder (root) because of these settings.

Example 3

Files Path Root

C:\source\root

Files

*

Options

DIVA Core will delete only the content of C:\source\root because of these settings.

Example 4

Files Path Root

C:\source\root

Files

*

Options

-delete_fpr

DIVA Core will delete only the content of C:\source\root, and eventually the parent folder (root) if it is empty, because of these settings.

Example 5

Files Path Root

C:\source\root

Files

object*

Options

-r

DIVA Core will delete the content of C:\source\root\object recursively and the parent folder (object) because of these settings.

Example 6

Files Path Root

C:\source\root

Files

object*

Options

-r -delete_fpr

DIVA Core will delete the content of C:\source\root\object recursively, then delete C:\source\root\object, and finally delete C:\source\root if it is empty because of these settings.

Example 7

Files Path Root

C:\source\root

Files

object1*
object2*

Options

-r

DIVA Core will delete the content of C:\source\root\object1 recursively, delete C:\source\root\object1, delete the content of C:\source\root\object2 recursively, and delete C:\source\root\object2 because of these settings.

Example 8

Files Path Root

C:\source\root

Files

object1*
object2*

Options

-r -delete_fpr

DIVA Core will delete the content of C:\source\root\object1 recursively, delete C:\source\root\object1, delete the content of C:\source\root\object2 recursively, delete C:\source\root\object2, and delete C:\source\root if it is empty because of these settings.

Example 9

Files Path Root

C:\source\root

Files

object1*
object2*

Options

-r -delete_fpr

DIVA Core will delete the content of C:\source\root\object1 recursively, delete C:\source\root\object1, delete C:\source\root\object2\subfolder\clip.mov, delete C:\source\root\object2\subfolder if it is empty, delete C:\source\root\object2 if it is empty, and delete C:\source\root if it is empty because of these settings.

Restore Requests

A *Restore* is defined as the transfer of an object to a Destination Server. You can initiate a Restore request from the Control GUI **Action** tab. Alternatively you can use the Objects view under the **Manage** tab by right-clicking the object to restore and selecting **Restore** from the resulting menu.

This request submits an Object Restore request to the Manager and the Manager chooses the appropriate instance to be restored. The request will fail if the requested object is on media that is not available.

The following fields are included in the Restore Request screen:

Object Name

The name of the object to be restored.

Note: Object Names cannot begin with a dollar sign ($).

Category

The category assigned to the object when it was archived. This parameter can be left empty but this may result in an error if several objects have the same name.

Instance

If multiple instances of an object reside in DIVA Core, you can specify which particular instance to restore. If left blank, DIVA Core will select the instance that provides the most optimum transfer.

Destination

Destination (*for example, a video server or browsing server*) for the object files. This name must be known by the DIVA Core configuration. Use the drop-down list to select the desired Destination.

Files Path Root

The root folder on the destination where the object files will be placed. This option appends or overrides the *FPR* used in the original archive request. If left empty, the files will be placed in the ***Files Path Root*** folder specified when archiving the object.

Options

Additional options for performing the transfer of data from DIVA Core to the Destination Server. These options supersede any options specified in the DIVA Core configuration database. Currently the possible values for *Options* are as follows:

No Entry
No entry in this field specifies no options.

-login
A user name and password is required to log in to some sources. This option obsoletes the **-gateway** option from earlier releases.

-pass
The password used with **-login**.

Quality of Service
One of the following codes (*see the* Quality of Service *section in this chapter for detailed descriptions*):

DIVA_QOS_DEFAULT
Restoring is performed according to the default Quality of Service (*currently direct and cache for restore operations*).

DIVA_QOS_CACHE_ONLY
Use cache restore only. Cache only restores can only restore from a tape instance. However, a tape instance on a tape in a group with a higher priority is preferred to a tape instance on a tape in a group with a lower priority.

DIVA_QOS_DIRECT_ONLY
Use direct restore only - no disk instance is created.

DIVA_QOS_CACHE_AND_DIRECT
Use cache restore if available, or direct restore if cache restore is not available.

DIVA_QOS_DIRECT_AND_CACHE
Use direct restore if available, or cache restore if direct restore is not available.

DIVA_QOS_NEARLINE_ONLY
Use Nearline restore only. Nearline restore will restore from a disk instance if one exists. Otherwise it will create a disk instance and restore from the newly created disk instance. However, a disk instance on a disk in an array with a higher priority is preferred to a disk instance on a disk in an array with a lower priority.

DIVA_QOS_NEARLINE_AND_DIRECT
Use Nearline restore if available, or direct restore if Nearline restore is not available. However, a disk instance on a disk in an array with a higher priority is preferred to a disk instance on a disk in an array with a lower priority.

Additional and optional services are available. To request those services use a logical OR between the previously documented Quality of Service parameter and the following constants:

DIVA_RESTORE_SERVICE_DO_NOT_OVERWRITE
Do not overwrite existing files on the destination server.

DIVA_RESTORE_SERVICE_DO_NOT_CHECK_EXISTENCE
Do not check for the existence of the clip on the server.

DIVA_RESTORE_SERVICE_DELETE_AND_WRITE
Force delete and rewrite if object exists on the server.

DIVA_RESTORE_SERVICE_DEFAULT
Operate using the default setting in the Manager configuration.

Priority

The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

  If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Source Media Priority

This value overrides the manager configuration parameter DIVAMANAGER_CACHE_QOS_USE_DISK, which prefers disk instances over tape instances during a Restore request.

Additional Services

Use the menu list to select whether DIVA Core will terminate the request if the file name on the destination already exists.

Archiving and Restoring in AXF Mode

In DIVA Core 8.1, an archive request for an AXF file results in DIVA Core automatically detecting that the file is an AXF file. Instead of archiving the AXF file itself, DIVA Core will archive the contents of the AXF file, retrieving the *Checksums* and *Provenance* of the object.

The restore request optional parameter **-axf** instructs DIVA Core to restore the original asset into an AXF file. Instead of purely restoring the content of an object to the destination, DIVA Core will restore the content into a new AXF Wrapper. When combined with **-rm** or **-rxml**, you can use this option to export an object with metadata information and then drop it into a DFM Watch Folder.

The AXF archive and restore functionality includes the following:

- Archive the content of an AXF file using auto-detection.
  - Identifies the .axf file name extension
  - Confirms it is a single file
  - Checks the beginning of the file for specific AXF properties
  - Checks for metadata information
- Restores an object into a new single AXF file. Previously this operation would have resulted in multiple files.
- Preservation of checksums
- Preservation of metadata
- Preservation of provenances
- Complex object support

This options works with **FTP_STANDARD**, **LOCAL**, **DISK**, **CIFS**, and **EXPEDAT** Servers.

Partial File Restore Requests

DIVA Core supports four types of Partial File Restore. The type of Partial File Restore implemented is determined by the ***format*** parameter in the request. This request submits a Partial Object Restore request to the Manager and the Manager chooses the appropriate instance to be restored. *If the requested object is on media that is not available the request will fail*.

The following list describes each type of Partial File Restore:

Files and Folders

This type of *Partial File Restore* enables extracting entire files from the archive or extracting entire directories and their contents. You can extract multiple files and directories in the same request. The files are restored with the file names and path names that were specified in the archive. There is no valid renaming option in *File and Folder Partial File Restore*. For example, a file archived as misc/12-2012/movie.avi will be partially restored to a misc/12-2012 subdirectory with the name movie.avi.

When a folder is specified in a *File and Folder Partial File Restore*, all files within that folder (*and the folder itself*) are restored. Additionally, each directory to be restored can include the **-r** option to recursively restore all folders nested within the target folder.

Byte Offset

This type enables extraction of a range of bytes from a particular file in the archive. For example, you can extract bytes 1 to 2000 (*the first 2000 bytes of the file*), or byte 5000 to the end of the file (*or both*) and store them to an output file such as movie.avi.

Note: The result of the *Byte Offset Partial File Restore* is usually not playable when applied to video files. The Datahub will not apply the header, footer, and so on, according to the video format.

Timecode

This type of Partial File Restore enables you to select a portion of a particular media file based on a timecode. For example, you could extract from 00:00:04:00 to 00:10:04:00 (*a 10 minute segment starting 4 seconds in and ending at 10 minutes and 4 seconds*), and place that segment into an output file such as movie.avi. The resulting file is a smaller version of the original movie file.

Note: The result of the Timecode Partial File Restore is a valid clip when applied to video files. The Datahub will apply the header, footer, and so on, according to the video format. If the Datahub cannot parse the format, the request will be terminated. This type of Partial File Restore can only be applied to a valid video clip.

DPX

This type of Partial File Restore enables extracting a range of DPX files from the archive. The entire object is viewed as a single media item, with one DPX file representing one frame of media. Only files with a .dpx, .tif, and .tiff extension in the archive are considered frames for the purposes of this command.

The first .dpx file (*or* .tif*, or* .tiff *file*) in the archived object is considered *frame 1*, the second .dpx in the archive is *frame 2*, and so on.

In the unlikely event that the .dpx, .tif, and/or .tiff files are mixed, the first sequential file of any of the three extensions will determine which files are considered to be part of the sequence. For example, if a stray .tif file is mixed with a collection of .dpx files and it came first in the sequence, the sequence is interpreted as a .tif sequence and .dpx files are ignored, even if this was not your intention.

For example, to extract frames 10 through 15 using *DPX Partial File Restore*, it restores the tenth .dpx file that appears in the archive, the eleventh .dpx file, and so on, ending with the fifteenth .dpx file, for a total of six files. Any other files (*such as* .wav *files*) are skipped by DPX Partial File Restore.

Special frame numbers 0 and -1 may be used to refer to the first and last frame respectively. *Frame 0* is valid as the start of a frame range and *Frame -1* is valid as the end of a range.

Valid frames and ranges are as follows:

- Frame 0 = first frame (*select the **Start of File** check box*)
- Frame 1 = the first frame in the sequence
- Frame n = the nth frame in the sequence
- Frame -1 = last frame (*select the **End of File** check box*)
- Specifying Frame 0 as the last frame is considered invalid.
- Specifying Frame 0 to 0 is currently invalid and will not return the first frame as might be intended.
- Specifying Frame 0 to 1 or Frame 1 to 1 will return the first frame.
- Specifying the Frame -1 in the first frame currently produces an error. You also cannot specify Frame -1 to -1 to return the exact last frame if the exact number of the last frame is unknown.

Examples:

startRange=0 - endRange=1
Restores only the first frame.

startRange=600 - endRange=635, startRange=679 - endRange=779
Restores frames 600 through 635, and frames 679 through 779.

startRange=810 - endRange=-1
Restores all frames from frame 810 to the end of the archive.

The actual file name may (*or may not*) match the frame number in DIVA Core. After restore DIVA Core interrogates the archive, finds the file order, and determines the *Frame Number* from the resulting file order found, it does not consider the file name. The first .dpx, .tif, or .tiff file found is considered *Frame 1*.

You must be careful when archiving DPX files to ensure they can be partially restored properly because *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 order in which the .dpx files appear in the archive. By default this order is based on ordering established by the source and is typically alphanumeric. For example, **NTFS DISK** Servers order files and folders case insensitively as a general rule, except where diacritical marks, such as ', `, ^, and so on are applied.

By default, when DIVA Core 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, but this can potentially create some issues. For example, if you 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.

DPX Partial File Restore looks at an entire object as a single piece of media. If multiple reels or clips appear in an archive, they can be stored in folders and partially restored using *File and Folder Partial File Restore*, but to *DPX Partial File Restore* they are viewed as one long movie clip. If this is a desired effect, you must ensure that the directories are sorted alphanumerically in the order the frames should be arranged.

DIVA Core does not perform any special audio handling for DPX media (*other than what might be embedded in DPX files*). DIVA Core can support transcoding of DPX media, but a transcoder may change the file names and/or file order of the DPX archive.

Submitting a Partial File Restore Request

You can submit a Partial File Restore request clicking on the **Partial Restore** button under the **Action** tab. Alternatively you can view the Archived Objects under the **Manager** tab, right-clicking the desired object and selecting **Partial Restore** from the resulting menu.

Either method results in the *Partial Restore Wizard* being displayed. If no object is selected and the **Partial Restore** icon (*under the **Action** tab*) is used, the wizard opens to **Step 1** (*of 3*) and the ***Object Name*** and ***Category*** must be entered manually.

If an object was selected and the (*right-click*) context menu was used, the wizard will open to **Step 1** (*of 2*). This step is similar to **Step 2** using the previous method to open the wizard window.

Use the following procedure to navigate through the wizard:

1. Enter the ***Object Name*** and ***Category***, or select the object in the left pane.
2. Click **Next** to proceed.
3. Select the type of *Partial File Restore* to perform using the menu list.

   Each type of *Partial File Restore* has different options associated with it, except for *Files and Folders Partial File Restore*, which does not have any specific options associated with it.
4. Drag-and-drop the objects from the left pane to the right pane to add them to the request.
5. Click **Next** to proceed.
6. You must include additional parameters for the following *Partial File Restore* types by double-clicking the object name after you moved it to the right pane.

   ### Byte Offset

   No offset is entered until you open the Options dialog box and manually enter one. Add the required ***Offset*** parameters and click **Add** to include them in the request.

   ### Timecode

   The ***File Format*** list is enabled after selecting the *Timecode Partial File Restore*. Select the proper file format from the drop-down list.

   Double-click the object to open the Options dialog box. Add the required ***Offset*** parameters and click **Add** to include them in the request.

   ### DPX

   Double-click the *DPX Frames* in the right pane to open the Options dialog box. Add the required ***Offset*** parameters and click **Add** to include them in the request.
7. After selecting the *Partial File Restore* type and associated options for each object, click the **Next** Button to go to the final screen.
8. Complete the required information on the final screen and click **Send** to send the request.

Note: Partial File Restore requests for AVI format files must include the same offset range (TCin, TCout) for all object components (*for example,* clip.avi*,* clip_1.wav*,* clip_2.wav).

The following list describes the parameters in the final Send Partial File Restore request screen:

Instance
If there are multiple instances of an object, DIVA Core will select the instance which will allow the request to complete in the least amount of time (*for example, a disk instance will be selected over a tape instance*). Specifying an instance number in this field will override this behavior and target the specific identified instance.

Destination
Destination (*for example, a video server or browsing server*) for the object files. This name must be known by the DIVA Core configuration. Use the drop-down list to select the desired Destination.

Files Path Root
The root folder on the destination where the object files will be placed. If left empty, the files will be placed in the ***Files Path Root*** folder specified when archiving the object.

Options
Additional options for performing the transfer of data from DIVA Core to the Destination Server. These options supersede any options specified in the DIVA Core configuration database. Currently the possible values for *Options* are as follows:

No Entry
No entry in this field specifies no options.

-login
A user name and password is required to log in to some Source Servers. This option obsoletes the **-gateway** option from earlier releases.

-pass
The password used with **-login**.

Quality of Service
One of the following codes (*see the* Quality of Service *section in this chapter for detailed descriptions*):

DIVA_QOS_DEFAULT
Restoring is performed according to the default Quality of Service (*currently direct for restore operations*).

DIVA_QOS_CACHE_ONLY
Use cache restore only.

DIVA_QOS_DIRECT_ONLY
Use direct restore only - no disk instance is created.

DIVA_QOS_CACHE_AND_DIRECT
Use cache restore if available, or direct restore if cache restore is not available.

DIVA_QOS_DIRECT_AND_CACHE
Use direct restore if available, or cache restore if direct restore is not available.

Additional and optional services are available. To request those services use a logical OR between the previously documented Quality of Service parameter and the following constants:

DIVA_RESTORE_SERVICE_DO_NOT_OVERWRITE
Do not overwrite existing files on the destination server.

Additional Services
Use the menu list to select whether DIVA Core will terminate the request if the file name on the destination already exists.

Priority
The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

  If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Multiple Restore (*N-Restore*) Requests

If an object is required on multiple destinations simultaneously, the Multiple Restore (*or N-Restore*) request enables specification of all necessary destinations in one command and submits it as a single request (*as opposed to multiple standard Restore requests for each destination*). This is also beneficial when the restore involves a tape instance because the tape is accessed once for the transfer rather than multiple read operations for single restore requests of the same object. Up to five simultaneous destinations are currently supported.

If the object to be restored is part of a spanned tape set, it must be restored to cache before the transfer to all destinations. If the transfer to one of the destinations fails, the others will still proceed (*if possible*) and the request status will be **Partially Aborted**.

If multiple renaming rules are defined DIVA Core will process the rule for each Server independently.

Use the following procedure to execute a Multiple Restore request:

1. On the **Action** tab of the ribbon bar, select the **Multiple Restore** button to open the Multiple Restore Request dialog box.
2. Enter the required parameters in the appropriate fields.

   To add multiple destinations, select the desired destination from the ***Destination*** menu list and click the double right-facing arrows to add the selected destination to the destination list text box field. Repeat this process until all required destinations are added to the list.

3. Click **Send** to process the request.

Delete and Delete Instance Requests

Use the **Delete** command to delete all instances of an object, or only a specific instance of the object from DIVA Core. *You must use this command with caution*. This command submits an object Delete request to the Manager and the Manager deletes every instance of the object (*unless otherwise specified*).

The ***Instance*** field of the Delete request determines exactly what will be deleted from DIVA Core. If this field is left empty, then all instances of that object will be deleted. A specific number entered into this field will only delete the specified instance.

You initiate the Delete request with the **Delete** button on the ribbon bar. You can also initiate it from the Objects view under the **Manage** tab by right-clicking the object to delete and selecting **Delete** from the resulting menu. If the **Delete** command is selected from the Objects view, the instance field is updated automatically with the selected instance. Only specific instance deletion is supported from this view.

Note: Deletes and repacks do not clear WORM media because these are Write-Once media. The instances are deleted but the space is not recoverable.

The following fields are included in the Send Delete Request screen:

Object Name
The name of the object to be deleted.

Note: Object Names cannot begin with a dollar sign ($).

Category
The category assigned to the object when it was archived. This parameter can be a null string, but this may result in an error if several objects have the same name.

Instance
If multiple instances of the object reside in DIVA Core, you can specify which specific instance to delete. If no number is entered in this field then DIVA Core will delete all instances of that object.

Media
The media can be an existing tape group or disk array. The drop-down list will only contain those items already configured in the DIVA Core configuration.

Priority
The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**

- **MAX**
- **DEFAULT**

  If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Deleting Instances on Cloned Tapes

Instances on tapes linked to a clone cannot be deleted. Attempts to delete an instance or object with instances on a Source or Clone Tape result in the request terminating. The clone link must be removed to delete the instance.

Copy Requests

Use the **Copy** command to create an instance of an existing object in the same or another group or array. This is useful for creating a backup copy of the object on another media.

Submits a request for copying an archived object to a new object, with another name and/or category, to the Manager and the Manager chooses the appropriate instance as the source of the copy. All types of transfers (*disk to disk, disk to tape, tape to disk, and tape to tape*) are supported.

In the event the requested object is on media that is not available, the request will fail.

When a Copy request is issued with no instance specified, and there are multiple instances of that object, DIVA Core will select the instance that will execute the copy operation in the shortest possible time (*for example, a disk instance will be selected over a tape instance*). If an instance number is entered in the ***Instance*** field of the request, the copy operation will use that specific instance only.

You initiate the Copy request with the **Copy** button on the ribbon bar, or the Objects View under the **Manage** tab by right-clicking the object to copy and selecting **Copy** from the resulting menu.

The following fields are included in the Copy Request screen:

Object Name

The name of the source object.

Note: Object Names cannot begin with a dollar sign ($).

Category

The category of the source object.

Instance

If multiple instances of the object reside in DIVA Core, you can identify which specific instance to copy. If no instance is specified, DIVA Core will select the instance that will provide the most optimal execution time.

Destination Media

The ***Destination Media*** can be either a tape group or disk array. If the new instance is being created in the same group or array, the request will only succeed if it can be copied to a separate tape or disk.

Priority

The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

  If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Copy As Requests

When an object is archived to DIVA Core, it is uniquely identified by its *Object Name* and *Object Category*. Neither the name nor the category can be altered once it exists within the DIVA Core database. The **Copy As** command allows creation of a new object in DIVA Core with a new *Object Name* and/or *Category*, and then the original object can be deleted if desired or necessary (*the latter must be performed manually*).

Note: Object Names cannot begin with a dollar sign ($).

You initiate the Copy As request with the **Copy As** button on the ribbon bar. You can also use the Objects View in the **Manage** tab and right-click the object, and select **Copy As** from the resulting menu.

For any object, the *Object Name* may not necessarily match that of the file name of the essence stored within it. If you use the **Copy As** command to create an object, it will still be restored using the same name with which it was originally archived.

Example:

If a file named xyz is archived from a server, regardless of what object name was given to it in DIVA Core, it will always restore to a destination as xyz regardless of its *Object Name*.

When a Copy As request is issued directly from the Objects View, the ***Instance*** field of the request is automatically left empty (*you can enter an instance number manually before the request is issued*). If this field is left empty when the request is submitted and there are multiple instances of that object, DIVA Core will select the instance that will complete the transfer in the shortest possible time by default (*that is, a disk instance will be selected over a tape instance*). This depends on the QOS specified and the DIVA Core configuration.

When the Copy As request is issued from the Object Properties View, the ***Instance*** field of the request is automatically updated with the number of the selected instance, and only that specific instance is copied. *Commands that are issued from the Objects Properties View must always specify an Instance Number*.

The following fields are included in the Copy As Request screen:

Source Object Name

The name of the source object.

Note: Object Names cannot begin with a dollar sign ($).

Source Object Category

The category of the source object.

Destination Object Name

The name of the destination object.

Note: Object Names cannot begin with a dollar sign ($).

Destination Object Category

The category of the destination object.

Instance

If multiple instances of the object reside in DIVA Core, you can identify which specific instance to copy. If no instance is specified, DIVA Core will select the instance that will provide the most optimal execution time.

Selecting ***Performance Optimized Instance*** instructs DIVA Core to use the instance that will achieve the request in the shortest time possible (*for example, a disk instance will be selected over a tape instance*).

Destination Media

The ***Destination Media*** can be either a tape group or disk array. If the new instance is being created in the same group or array, the request will only succeed if it can be copied to a separate tape or disk.

Destination Storage Plan

The ***Storage Plan*** to assign the new object on the destination.

Comments

Comments added here will be added to the new object's properties.

Priority

The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Associative Copy Requests

The Associative Copy request works with the Archived Objects View and enables you to copy multiple objects sequentially to a single tape in a specified group. An example is backing up the selected objects to a single tape so it can subsequently be externalized.

You initiate the Associative Copy request with the **Associative Copy** button on the ribbon bar. Alternatively, you can use the Objects View under the **Manage** tab by right-clicking the object to copy and selecting **Associative Copy** from the resulting menu.

The objects available for an Associative Copy request must first be obtained by executing a query in the Archive Objects view. You can target specific object by name, category, and/or creation date.

Associative copying involves reading and writing files from the source group(*s*) to the destination group one file at a time. DIVA Core guarantees that these instances are stored sequentially on tapes with the following exceptions:

- It is not compatible with tape spanning. If no tape is currently available for copying all of the selected objects to a single tape, the request terminates (*and is retried once*) instead of spanning. If the sum of the size of the objects to copy exceeds the capacity of every individual tape present in the library, the request terminates.
- Two or more instances of an object on the same tape are not permitted. This may reduce the range of tapes that can be selected for the Associative Copy. If no appropriate tape is available to meet this condition, the request will be terminated.
- The request is complete only when every object has been copied onto the same tape.
- If a drive or tape failure during a write operation occurs, instances currently written are erased and the request is retried once.
- Choice of the tape to be used for the copy follows the policy used for the archive operation (*written tapes with enough remaining size regardless of optimizations*).

The following fields are included in the Associative Copy Request screen:

Main Display Field

All objects returned from the query in the Archived Objects view are displayed in the Associative Copy request. However, only those selected when the command was issued are highlighted. The highlighted entries can be subsequently selected or deselected by using either the **CTL** or **SHIFT** keys in combination with the mouse.

Destination Group

Only tape groups currently configured in DIVA Command will be displayed in this list. You must select your destination group from the list.

Priority

The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**

- **HIGH**
- **MAX**
- **DEFAULT**

  If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Require and Release Requests

Requiring or releasing objects is primarily a database entry for instances that are, or can be, externalized from the tape library. Tapes that can be externalized or must be internalized are determined through the Require Instances view.

Require indicates to the Manager that this instance must be inserted. This request has no effect if the instance is already required. You can retrieve a list of instances that are **REQUIRED** and **EJECTED** from the Control GUI.

You initiate the Require or Release request with the **Require/Release** button on the ribbon bar. You can also use the Objects view under the **Manage** tab and right-click the desired object and selecting either **Require** or **Release** from the resulting menu.

The following fields are included in the Send Require Request screen:

Object Name
The name of the required object.

Note: Object Names cannot begin with a dollar sign ($).

Category
The category assigned to the object when it was archived. This parameter can be left empty but this may result in an error if several objects have the same name.

Instance
No value entered here forces the function to apply to every instance of the given object.

Release indicates to the Manager that this instance can be externalized. This request has no effect if the instance has already been released. You can retrieve a list of instances that are **RELEASED** and **INSERTED** from the Control GUI. A releasable tape is one that only contains released instances.

The following fields are included in the Send Release Request screen:

Object Name
The name of the required object.

Note: Object Names cannot begin with a dollar sign ($).

Category
The category assigned to the object when it was archived. This parameter can be left empty but this may result in an error if several objects have the same name.

Instance
No value entered here forces the function to apply to every instance of the given object.

Eject Tape Requests

The Eject Tape request ejects the selected tapes from the associated library. You can select one or more tapes simultaneously. You initiate the Eject Tape request with the **Eject Tape** button on the ribbon bar, or in the Tapes View under the **Home** tab by right-clicking the tape to eject and selecting **Eject Tape** from the resulting menu.

The following fields are included in the Eject Tape Request screen:

Comments
Comments can be added when the tape is ejected. These may refer to its location or other information. you can view comments later by examining that tapes properties in the Tapes View.

Release instances on tape(s)
When selected all object instances on the tape being ejected are released.

Priority
The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

  If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Ejecting Cloned Tapes

Ejecting Cloned Tapes works in the same way as exporting them; the associated tape will be ejected as well. To disable this behavior, remove the clone link. Then, if you only eject the Clone Tape, you must set the Source Tape to not writable.

Insert Tape Requests

This request enables inserting a tape into a library through its CAP. You can only enter the tapes in the CAP after this command is issued with some library configurations.

Note: Contact Telestream Support for instructions on bulk loading of tapes into a library.

You initiate an Insert Tape request using the **Insert Tape** button located under the **Action** tab on the ribbon bar, and then the **Tape Actions** button.

The Sony PetaSite PSC software enables you to enter a tape in its CAP and manually place it within the PetaSite. In this case, DIVA Core is not informed of the action by the PSC and will not recognize the tape until the library is audited using DIVA Command.

The following fields are included in the Insert Tape Request screen:

Require instances on tape

When selected, any *Released* instances on the inserted tape are set to *Required*.

Robot Manager Name

This list specifies the Robot Manager controlling the associated library for insertion of the tapes.

CAP ID

This list is for libraries with multiple CAPs. Some libraries will not unlock the CAP, enabling the tape to be inserted, until the **Insert Tape** command is issued. You can specify which CAP to unlock from this list.

Priority

The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

  If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Repack Tape Requests

The Repack Tape request sends a repack request for the selected or specified tape. A tape repack operation reclaims unusable space on a tape due to object deletions, and removes fragmentation.

Caution: The repack tape function is not intended to move material from a tape that is already known to be generating read errors. *Contact Telestream Support for advice in these situations*.

You initiate the Repack Tape request with the **Repack Tape** button on the ribbon bar, or the Tapes View under the **Home** tab by right-clicking the tape to repack and selecting **Repack Tape** from the resulting menu.

Tape repacking can be a lengthy process and DIVA Core, by default, considers tape repacks a low priority operation. If higher priority requests are issued a tape repack request can be temporarily suspended while the higher priority requests are completed. If higher priority requests are issued sporadically it can result in frequent mount and dismount operations of the drive performing the repack. Therefore, Telestream recommends that tape repack operations should be run during off-peak periods where the frequency of higher priority requests is limited. Some installations may have a drive dedicated solely to tape repacking to prevent this scenario from occurring.

The repack process involves the following tasks (*in order*):

1. Mounting the source tape and reading all objects to temporary disk cache of an Datahub enabled for repack operations.

> **Note:** If the temporary disk cache is filled before reading all objects from the source tape, DIVA Core will begin proceed to Step 2 until the cache is cleared. DIVA Core will then proceed to read the remaining objects from the source tape. This process is repeated until all objects are read.

2. Mounting a tape from the *Unused Tapes Sets* pool associated with the Set ID of the group from the source tape.
3. Writing all objects from the temporary cache in Step 1.
4. Deleting the objects from the temporary cache after all objects have been successfully written to the new tape.
5. The original source tape is released to the *Unused Tapes Sets* pool and unassigned from the group.

If a read error occurs at some point during the repack process from the source tape or a write error occurs on the destination tape, the entire repack request is terminated and no objects from the source tape are deleted. If the cache filled during the repack request and objects were successfully written to another tape before the cache was cleared, those objects will remain on the destination tape.

If a read error occurred, the source tape will have the repack status and write status disabled. If a write error occurred, the destination tape will have its write status disabled and will not be used for any tape write operations. You can view the write and repack states of both tapes in the Tape States frame of DIVA Command.

During the manual repack of WORM media, the usual dialogs display, but a warning is included notifying you that the space on the source media will not be recoverable after the repack is complete. Deletes and Repacks do not clear WORM media because these are Write-Once media. The instances are deleted, but the space is not recoverable.

You initiate the Repack Tape request with the **Repack Tape** button on the ribbon bar, or the Tapes View under the **Home** tab by right-clicking the tape to repack and selecting **Repack Tape** from the resulting menu.

The following fields are included in the Repack Tape Request screen:

Repack WORM (Write-Once media) with barcode

Barcode of the tape to be repacked.

Priority

The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Repacking Cloned Tapes

Tapes linked to a clone cannot be repacked. If you attempt to repack a Source or Clone Tape the request will terminate. The clone link must be removed to repack either tape.

Verify Tape Requests

The Verify Tape request initiates a system read-back through every object on the selected tape one at a time and verifies all of the checksum values.

You initiate the Verify Tape request with the **Verify Tape** button on the ribbon bar, or the Tapes View under the **Home** Tab by right-clicking the tape to verify and selecting **Verify Tape** from the resulting menu.

The following fields are included in the Verify Tape Request screen:

Verify Tape with barcode

Barcode of the tape to be verified.

Priority

The priority level for this request. The level can be in the range zero to one hundred, or the value **DEFAULT**. The value zero is the lowest priority and one hundred the highest priority. Move the slide control to increase or decrease the request priority.

There are six predefined values as follows:

- **MIN**
- **LOW**
- **NORMAL**
- **HIGH**
- **MAX**
- **DEFAULT**

If the **DEFAULT** check box is selected, the slide control becomes inactive and the priority defined in the Manager configuration is used.

Export and Import Tape Requests

The Export Tape request type enables one or more tapes containing objects to be exported to another independent DIVA Core platform (*for example, at a remote disaster recovery or partner site*).

Note: *See Chapter 7, "Removable Media Operations"for details on tape exporting and importing, bulk tape exporting and importing, encrypted tape exporting and importing, and API exporting and importing functionality*.

The metadata of each tape (*that is, the object names and categories it contains and their location on the tape itself*) are maintained in the DIVA Core Oracle Database. For complex objects this information is also in the Metadata Database. Additionally, the metadata of each tape is saved to an XML file when the tapes are exported. The XML file transfers the metadata of each tape into the other DIVA Core platform's database when the tapes are imported.

The **Export Tapes** command is *not* used for transferring tapes between two or more libraries controlled by the same Manager. Tapes (*and the instances they contain*) exported from DIVA Core using this command are also removed from the DIVA Core database. If the object being exported is the last (*or only*) instance of that object, it will be removed entirely from the database.

The following new parameters have been added for export and import tape functions. *All XML metadata files exported from all previous DIVA Core versions will not continue to be supported.*

During an export and import of WORM media, whether the media is Write-Once and the media is a cartridge is identified in the exported XML file. This information is imported with the attributes isWriteOnce and isCartridge being either true or false.

Importing WORM media is supported by DIVA Core 7.4 and later. When you import DIVA Core 7.4 (*or later*) WORM media into a DIVA Core release earlier than DIVA Core 7.4, DIVA Core ignores the WORM flag (*it is set to* false), and logged in the Manager log. The device will be visible in the Control GUI as a tape but unusable if finalized, or if no WORM drive is connected to the system.

The following table describes the export and import parameters:

Table 2–1 Export and Import Parameters

Parameter	XML Element or Attribute	Notes
objectId	Attribute of the object element	Not imported. A new Object ID is generated during import.
uuid	Attribute of the object element	Imported if present, otherwise a new UUID will be generated.
numFolders	Attribute of the object element	
format	Attribute of the object element and attribute of the tape element	0 = legacy 1 = AXF -1 = unknown
numFolders	Attribute of the object element	
isHeaderValid	Attribute of the object element	
isComplex	Attribute of the object element	
footerBeginPos	Attribute of the element	If exists in database
footerEndPos	Attribute of the element	If exists in database
compOrderNumBegin	Attribute of the element	If exists in database
compOrderNumEnd	Attribute of the element	If exists in database
fileFolderMetadataInfo	Element	Valid for complex objects
fileFolderMetadataInfo-elem	Element	Valid for complex objects
checksums and checksum	Element	Not valid for complex objects

You initiate the Export Tape request with the **Export Tape** button on the ribbon bar, or the Tapes View under the **Home** tab by right-clicking the tape to export and selecting **Export Tape** from the resulting menu.

The following fields are included in the Eject Tape Request screen:

Comments
Enter any desired comments in this field.

Delete from DB
When selected removes selected tape(*s*) from the *Exported Tapes* list.

Exported Tapes
This area displays tapes selected for export. These tapes and the instances they contain will be removed from the DIVA Core Database once exported. If required, select any tapes to be removed from the *Exported Tapes* list and click **Remove Selected**.

Exported Tape Metadata Files

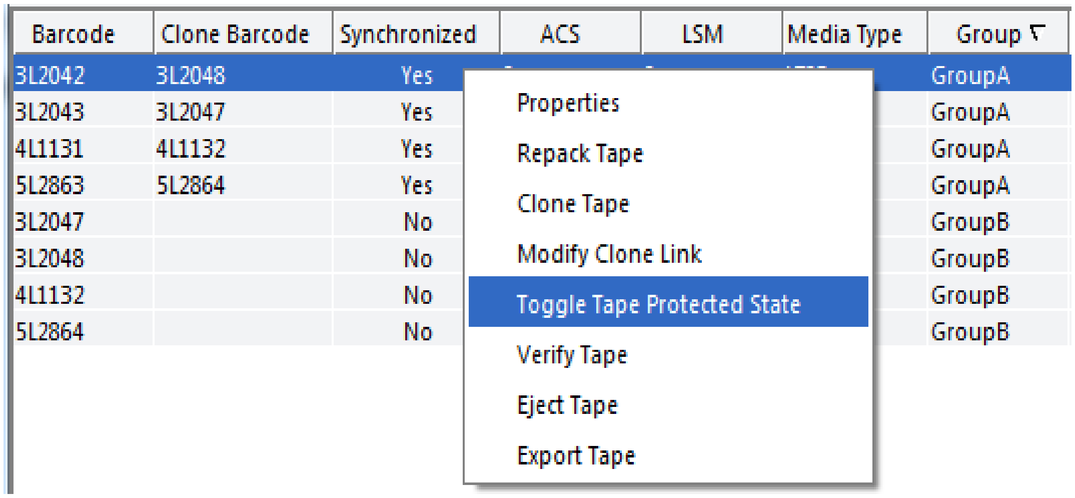
DIVA Core writes each tape's metadata to an XML file when the tapes are exported from your system. If an object is spanned across two (*or more*) tapes, the XML file will encompass every tape in the spanned set. The naming format of each tape metadata XML file is Tapeset-<Barcode>.xml (*for example, Tapeset-000131.xml*).

The root path where the XML files are saved is defined by the DIVAMANAGER_EXPORT_ROOT_DIR parameter in the Manager configuration file (*consult your Site Configuration for these details*). By default, the export absolute directory root path is %DIVA_HOME%\Program\Manager\bin\exported\. From this root path, the XML files from each **Export Tapes** command are saved in subfolders based on the date and time the command was run.

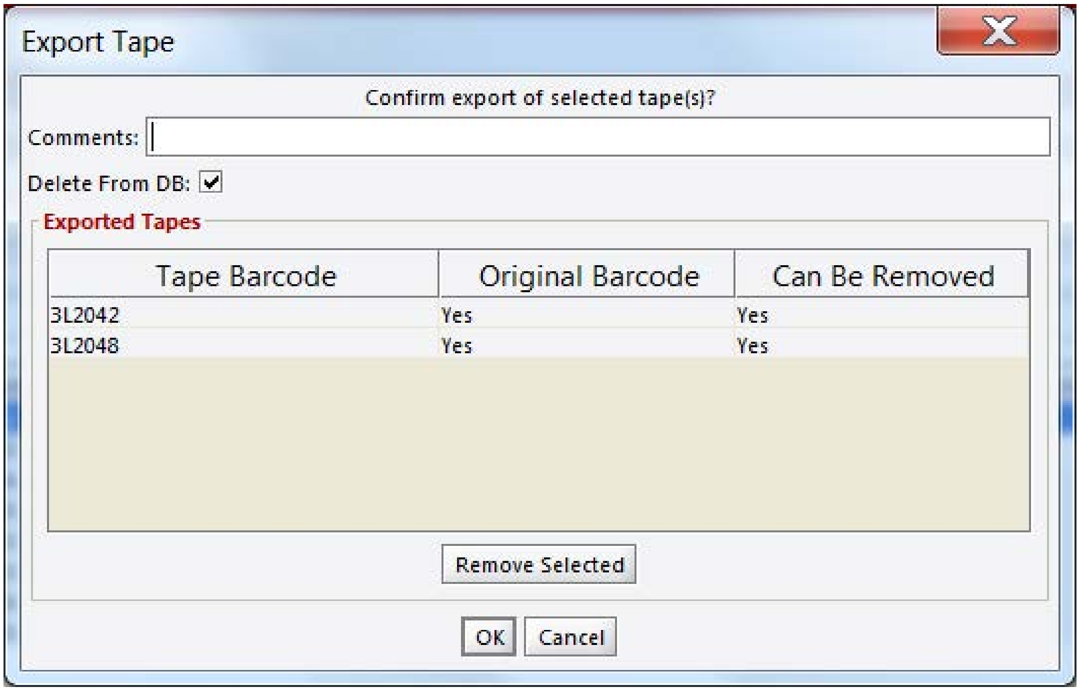
Exporting Cloned Tapes

Exporting a Source or Clone Tape also triggers the export of the associated tape. In a previous example, content was archived to Source Tape 3L2042, and then the contents were cloned to tape 3L2048. When exporting Source Tape 3L2042, Clone Tape 3L2048 will automatically be included in the list of tapes to export. Similarly, exporting Clone Tape 3L2048 will trigger the export of tape 3L2042.

To remove this association and not export the associated tape, select **Modify Clone Link** from the set of tape options available when you right-click a tape and clear the barcode. Also, on exclusive export of the Clone Tape, you must mark the Source Tape as *Protected* by selecting the **Toggle Tape Protected State** option.



Barcode	Clone Barcode	Synchronized	ACS	LSM	Media Type	Group
3L2042	3L2048	Yes				GroupA
3L2043	3L2047	Yes				GroupA
4L1131	4L1132	Yes				GroupA
5L2863	5L2864	Yes				GroupA
3L2047		No				GroupB
3L2048		No				GroupB
4L1132		No				GroupB
5L2864		No				GroupB



Export Tape

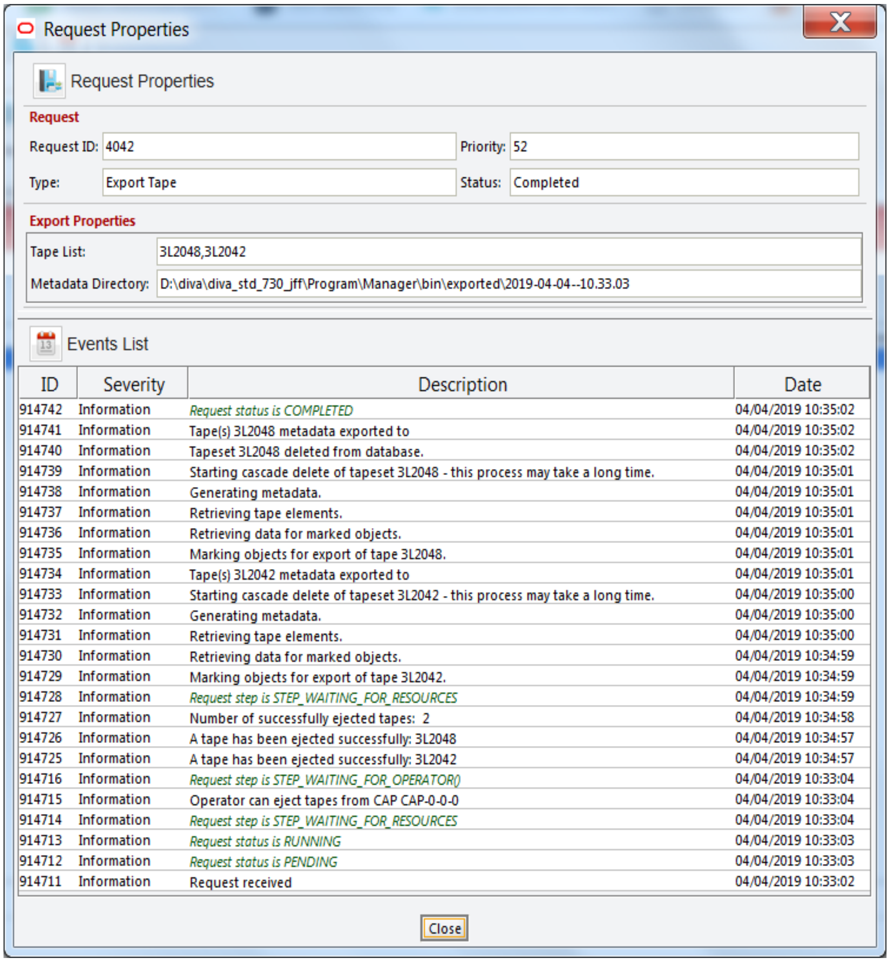
Confirm export of selected tape(s)?

Comments:

Delete From DB: ☑

Exported Tapes

Tape Barcode	Original Barcode	Can Be Removed
3L2042	Yes	Yes
3L2048	Yes	Yes



Request Properties

Request Properties

Request

Request ID: 4042

Priority: 52

Type: Export Tape

Status: Completed

Export Properties

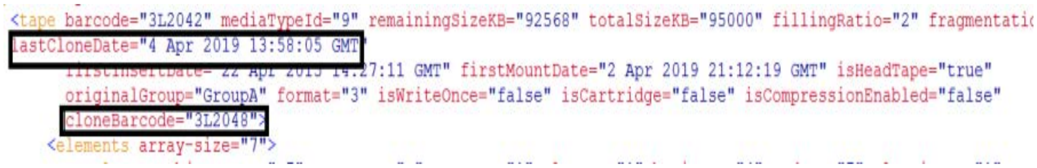
Tape List: 3L2048,3L2042

Metadata Directory: D:\diva\diva_std_730_jff\Program\Manager\bin\exported\2019-04-04--10.33.03

Events List

ID	Severity	Description	Date
914742	Information	*Request status is COMPLETED*	04/04/2019 10:35:02
914741	Information	Tape(s) 3L2048 metadata exported to	04/04/2019 10:35:02
914740	Information	Tapeset 3L2048 deleted from database.	04/04/2019 10:35:02
914739	Information	Starting cascade delete of tapeset 3L2048 - this process may take a long time.	04/04/2019 10:35:01
914738	Information	Generating metadata.	04/04/2019 10:35:01
914737	Information	Retrieving tape elements.	04/04/2019 10:35:01
914736	Information	Retrieving data for marked objects.	04/04/2019 10:35:01
914735	Information	Marking objects for export of tape 3L2048.	04/04/2019 10:35:01
914734	Information	Tape(s) 3L2042 metadata exported to	04/04/2019 10:35:01
914733	Information	Starting cascade delete of tapeset 3L2042 - this process may take a long time.	04/04/2019 10:35:00
914732	Information	Generating metadata.	04/04/2019 10:35:00
914731	Information	Retrieving tape elements.	04/04/2019 10:35:00
914730	Information	Retrieving data for marked objects.	04/04/2019 10:34:59
914729	Information	Marking objects for export of tape 3L2042.	04/04/2019 10:34:59
914728	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	04/04/2019 10:34:59
914727	Information	Number of successfully ejected tapes: 2	04/04/2019 10:34:58
914726	Information	A tape has been ejected successfully: 3L2048	04/04/2019 10:34:57
914725	Information	A tape has been ejected successfully: 3L2042	04/04/2019 10:34:57
914716	Information	*Request step is STEP_WAITING_FOR_OPERATOR()*	04/04/2019 10:33:04
914715	Information	Operator can eject tapes from CAP CAP-0-0-0	04/04/2019 10:33:04
914714	Information	*Request step is STEP_WAITING_FOR_RESOURCES*	04/04/2019 10:33:04
914713	Information	*Request status is RUNNING*	04/04/2019 10:33:03
914712	Information	*Request status is PENDING*	04/04/2019 10:33:03
914711	Information	Request received	04/04/2019 10:33:02

The resulting exported XML file(*s*) contains a new element named cloneBarcode. It is used during import to restore the clone link. It also has the lastCloneDate for the Source Tape that is also restored during import.



```
<tape barcode="3L2042" mediaTypeId="9" remainingSizeKB="92568" totalSizeKB="95000" fillingRatio="2" fragmentati
lastCloneDate="4 Apr 2019 13:58:05 GMT"
    firstInsertDate="22 Apr 2019 14:27:11 GMT" firstMountDate="2 Apr 2019 21:12:19 GMT" isHeadTape="true"
    originalGroup="GroupA" format="3" isWriteOnce="false" isCartridge="false" isCompressionEnabled="false"
    cloneBarcode="3L2048">
  <elements array-size="7">
```

Tape Import Workflow

To use the importtapes command, first ensure that the XML metadata file and the .ffm files that were exported exist on the DIVA Core system into which they are to be imported. The files must exist in uncompressed form (*unzipped*) in the Manager's bin directory (*by default*). The object tape group must also already exist on the target system before the import begins. This tape group does not necessarily have to be the same group that the tape was assigned to in the source system.

There are three main ways that a tape object can be treated during the import process:

- Imported as a new object
- Skipped if the object is already present in the system
- Imported as an instance of a pre-existing (*in the DIVA Core database*) object. *This option only functions correctly if the checksums match*.

See the Installation and Configuration Guide for more information.

Importing Tapes

The associated XML files from exported tapes must be copied to the Manager bin directory on the platform where they are to be imported. Before inserting the tapes into the library, you must import the tape metadata into the DIVA Core database. You must execute this for each tape (*or spanned set*) to be imported.

1. Open a command prompt window.
2. Execute the following commands in the order shown:

   ```
   cd \%DIVA_HOME%\Program\Manager\bin
   importtapes <destination_group> <metadata_file>
   ```

   destination_group
   The tape group that the tape (*or spanned set*) and its instances will be assigned to in the destination system.

   metadata_file
   The file name of the XML file for the tape (*or spanned set*).

Example:

The tape with barcode number 000131 also contains objects that are spanned across the tape with a barcode of 000120. When tape 000131 is exported, the exported XML file is named Tapeset-000131.xml. This .xml file also encompass the objects from tape 000120, and both tapes are ejected from the library.

After all objects from both tapes are exported to the XML file, all instances on each tape (*and references to the tapes themselves*) are removed from the DIVA Core database. The XML file is then copied to the %DIVA_HOME%\Program\Manager\bin folder of the destination DIVA Core system where it is to be imported.

You import the metadata for this tape into the group MOVIES using the following command:

```
importtapes MOVIES Tapeset-000131.xml
```

After the tape's metadata successfully imports into the database (*check the Control GUI Current Requests queue*), both tapes and their objects are considered externalized and can both be entered into the library with the **Insert Tape** command.

Importing Cloned Tapes

To restore the unidirectional link with the same Source and Clone Tape designations, you must import the clone link first. Otherwise, the relationship becomes reversed; that is, the Source Tape becomes the Clone Tape and vice-versa. This is not critical because these tapes are clones of each other. For example:

```
ImportTapes "GroupB" "D:\diva\diva_std_730_
jff\Program\Manager\bin\exported\2019-04-04--10.33.03\tapeset-3L2048.xml"

The import completed successfully. (Code 0)
```

Importing the associated tape requires the use of the -addAsInstanceIfNameExists option, because the same objects will be imported from the associated tape. For example:

```
ImportTapes addAsInstanceIfNameExists "GroupA" "D:\diva\diva_std_730_
jff\Program\Manager\bin\exported\2019-04-04-10.33.03\tapeset-3L2042.xml"

The following objects to be imported have the same Object name and Category as Objects already in the DIVA database...
1. A19/A 2. A18/A 3. A17/A 4. A16/A 5. A7/A 6. A21/A 7. A20/AW A R N I N G *** The 7 object names listed above exist on
the tape to be imported, but ALSO exist in the DIVA system you are importing TO. * You have chosen to first COMPARE
the component CHECKSUMS of the objects that have naming conflicts. If that match succeeds, these objects will be
imported as instances of the objects they match to. * If the checksums match and you import as an instance, you may lose
the Comments, UUID, Archive Date and/or archived path root that were stored in the import metadata. * If they don't
match, the import will fail, and then will exit...

The import completed successfully. (Code 0)
```

Migrate Content Requests

The Migrate request enables tape content migration to another tape group or disk array. For example, if you are upgrading to a new media type in your library and want to move the content from the old legacy tapes to the new AXF format.

Note: DIVA Core includes an embedded migration service (*DIVAmigrate*). It is a separate internal (*to DIVA Core*) service which helps users to schedule and run jobs to migrate content between different media inside of a DIVA Core system. You can use the Control GUI or command line client. *Contact Telestream Support for more details*.

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.

This request type is only available from the Control GUI Tapes View. It also uses the SPM (*Storage Plan Manager)* option to perform the migration (*SPM must be installed*). Appropriate slots must be configured for the migration in the DIVA Command **Slots Tab** in the Storage Plans screen. The slots indicate to DIVA Core when the migration operations are to be performed

See the SPM (Storage Plan Manager) Guide for more information.

A Migrate request will perform the following functions (*in the order shown*):

1. Mount the source tapes and issue Copy requests to DIVA Core to copy from the source tape to the destination media (*disk or tape*). Any objects that are spanned will require the object to be copied first into cache.
2. Delete the Source Server instance after the object has been successfully copied to the new media.
3. The source tapes are cleared of all objects and returned to the *Unused Tapes Sets* pool.

You initiate the Migrate request with the **Migrate** button on the ribbon bar, or the Tapes View under the **Home** tab by right-clicking the tape to migrate the content from and selecting **Migrate Content** from the resulting menu.

The following fields are included in the Migrate Content Request screen:

Task Name
Enter the name of the migration task in the field.

Migrate content from tapes
This list identifies the barcodes of tapes selected for migration.

Migrate Slot
Select the migration slot you want to use from the list. The slot is defined in the **Slots** tab of the **Storage Plans** menu item in DIVA Command. The slots dictate when the migration requests will be issued to DIVA Core.

Dest. Medium
The destination medium (*tape group or disk array*) for migrated content.

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 under the **Job Events** tab highlights events using different colors based on severity. Red indicates an *Error*, yellow indicates a *Warning*, and white indicates *Information*. Clicking the **Refresh** button refreshes the entire Job Properties dialog box.

Metasources

The Metasource source type allows several currently defined DIVA Core Servers sharing the same online storage (*or monitoring the same folder or FTP server for Drop Folder Monitors*) to be combined and considered a single DIVA Core Server configuration. This unique (*and optional*) feature enables DIVA Core to provide automatic load balancing and failover capabilities in case of one or more of the individual Servers going offline.

When requests are issued to DIVA Core with a Server using the Metasource source type, each additional Archive or Restore request will use the next server in the Metasource list. If the server selected by DIVA Core is offline or encounters an error, DIVA Core will automatically attempt to use the next server in the Metasource list. If all servers fail to fulfill the transfer, the request will stop.

Symbolic Links

You can archive and restore symbolic links on Linux in DIVA Core 8.1. Symbolic links are only supported for the **AXF** format. When using **LEGACY** format DIVA Core reports an error if a symbolic link is discovered during the transfer.

Symbolic Links are only supported with an SFTP Server on Windows. You must specify the following options when configuring SFTP:

```
-login [login] -pass [password] -port 22 -socket_block_size 64
```

DIVA Core running under Linux supports **CIFS**, **DISK**, and **LOCAL** Source and Destination Servers.

When restoring an object containing symbolic links to a destination server that does not support them they are ignored and not created on the destination server.

Symbolic links created using the Windows operating system are not supported. Shortcuts created using the Windows operating system are not represented as symbolic links because they are treated as files. Only symbolic links created on UNIX platforms are archived and represented as symbolic links in DIVA Core.

In the Control GUI, the file type displays in the *Components* list on the **Properties** tab of the Object Properties window. The possible types are **File** (**F**), **Directory** (**D**) and **Symbolic Link** (**S**). Symbolic links are also displayed in the *Elements* list on the **Instances** tab of the Object Properties window, and as files under the *Components* tab of the Object Properties screen.

Storage Plan Management

Caution: Misconfiguration of SPM (*Storage Plan Manager*) 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 Telestream Support before making any changes to SPM. Failure to do so may result in severe damage to the DIVA Core system or even permanent data loss.

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

The newest release of SPM supports disk cleaning based on the object's archived date. Earlier releases of SPM's disk cleaning feature only supported cleaning based on an object's last access time and object size.

Content Verification

The purpose of the Checksum Support and Content Verification program is to introduce additional levels of verification into the DIVA Core system. This feature introduces checksum generation and verification for each file managed by DIVA Core. The currently supported checksum algorithms in DIVA Core include **MD2**, **MDC2**, **MD5**, **SHA**, **SHA-1**, and **RIPEMD160**.

Note: Additional checksum verification is performed at the Oracle Storage Cloud level. See the *Storage Cloud documentation* for information.

The default and recommended checksum algorithm is **MD5**. Although the other algorithms are maintained for backward compatibility, only **MD5** and **SHA-1** are recommended for best results.

When an object contains multiple files a checksum will be generated and later verified for each of the component elements. The following three types of checksum sources are distinguishable:

- Genuine Checksum
- Archive Checksum
- Deferred Checksum

The *TEXT Genuine Checksum* mode enables DIVA Core to archive all files and subfolders in a specified folder while comparing their checksum values against known values stored in an external checksum file. Files that do not have a matching checksum in the external checksum file are archived with DIVA Core's calculated checksum and the external checksum file will not be archived.

Note: The *TEXT Genuine Checksum* is a customer-specific implementation and only supports **MD5**. Unicode is not supported, and checksums must be in a .md5 text file.

Archive Instructions

Use the following procedure to archive objects using checksum verification through the Control GUI:

1. In the Manager Control GUI, navigate to **Action** and then select **Archive**.
2. From the ***Source*** drop-down list, select the Server entry that was created in the configuration stage.
3. Enter the desired ***File Path Root*** in the ***File Path Root*** text field.
4. Enter the path to the location of the checksum files in the ***Files*** field and append your entry with a wildcard symbol (*an asterisk*).
5. Enter -r In the ***Options*** field.
6. Enter the remaining parameters in the request form and click **Send**.

Limitations

The following limitations apply when using checksum verification. *See the Checksum Support User's Guide for additional information*.

- DIVA Core cannot open or create files on a Windows file system if their absolute path exceeds 256 characters. The *Root Path* must be no more than a total of 256 characters.
- Only ASCII, non-UTF-8 encoded checksum files are supported.
- Each line in the checksum file must begin with an MD5 checksum, followed by 2 spaces, and then the *File Path* to the referenced file.

Genuine Checksum using AXF Transfer

The AXF Genuine Checksum mode enables DIVA Core 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 kind of workflow is typically combined with a Restore request with -axf in the ***Request Options*** field.

Requirements

The AXF object containing the files to be archived must contain checksum information for each file. The supplied checksum in the AXF object must be the expected type as defined in the configuration.

DIVA Core DIVA Command Settings

The DIVA Core settings are located under the **DIVA Configuration** menu item in DIVA Command.

1. Open DIVA Command and navigate to the **DIVA Configuration** menu item (*on the left of the screen*).
2. Select the **System** sub-menu item.
3. Click the plus icon (**+**) at the top of the **Servers** tab to create a new ***Server*** entry.
4. Fill n the fields with ***Source Type*** set to either **DISK**, **FTP_STANDARD**, or **EXPEDAT** as appropriate.
5. Specify an appropriate ***Root Path*** if necessary. This path, along with the input files specified during the Archive request, determines the location of the checksum file.

   For example, if the ***Source Type*** is **DISK** the ***Root Path*** can be D:\root. If the ***Source Type*** is **FTP_STANDARD**, the ***Root Path*** can be /root.
6. Set the ***External Checksum Source*** to **YES**.
7. Set the ***Checksum Type*** to the expected checksum type (*for example, MD5*).
8. Set the ***GC Mode*** to **AXF**.
9. Click **OK**.
10. Select **Tools**, and then **Notify Manager** from the menu to notify the Manager of the configuration.

Archive Instructions

Use the following procedure to archive an object with Genuine Checksum through an AXF transfer:

1. In the Manager Control GUI, navigate to **Action** and then select **Archive**.
2. From the ***Source*** drop-down list, select the Server entry that you created in the Configuration procedure.
3. Set the desired ***File Path Root***.
4. Enter in the path to the location of the AXF file in the ***Files*** field. The file extension must be.axf.
5. Enter the remaining parameters in the request form and click **Send**.

Limitations

The workflow described only works with AXF requests generated by DIVA Core.

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. The -axf option does not create an actual restore, but rather an object export in an AXF Wrapper. These options are mutually exclusive and should not be part of the same workflow.

See the *Checksum Support User's Guide* for more information.

Quality of Service

The QOS (***Quality of Service)*** parameter defines how a file is transferred to and from a tape, from a source, or to a destination. The following ***Request Options*** map to their logical quality of service:

```
-qos_direct_only
-qos_cache_only
-qos_direct_and_cache
-qos_cache_and_direct
-qos_nearline_only
-qos_nearline_and_direct
```

Request Options take precedence over the normal quality of service specification. Also, the normal ***Quality of Service*** specification takes precedence over the ***Server Connect Options***.

NEARLINE_ONLY and **NEARLINE_AND_DIRECT** QOS values are now supported in the ***Server Connect Options***. These options are only valid for a Restore request. DIVA Core ignores the setting and the usual default is applied if a Source or Destination Server with either setting is used in any other type of request. The QOS value is no longer case-sensitive and no longer must be specified at the beginning of the options.

For example, `-login test -pass test qos=nearline_only` is a valid option.

The options for QOS are defined as follows:

Direct Only (-qos_direct_only)

The data is transferred immediately to the source as it is read from tape. Alternately, DIVA Core writes the data to tape immediately as it is transferred from a destination. If no direct transfer service is available, the request terminates.

Cache Only (-qos_cache_only)

The data is first transferred entirely to cache storage from tape, and then transferred to the destination. Alternately, the data is first transferred entirely from the source to cache storage, and then written to tape. If no cache service is available, the request terminates.

Direct and Cache (-qos_direct_and_cache)

If a direct transfer is not available, for example no Datahub with direct transfer enabled is available, then a cache transfer is performed.

Cache and Direct (-qos_cache_and_direct)

If cache transfer is not available, for example no Datahub with cache storage is available, then a direct transfer is performed.

Nearline Only (-qos_nearline_only)

This is only available for Restore and N-Restore requests. If a Nearline disk instance exists, the data is transferred from Nearline disk to the destination. Alternatively, the data is first transferred entirely to Nearline storage on disk from tape, and then transferred to the destination. If no Nearline service is available, the request terminates.

Nearline and Direct (-qos_nearline_and_direct)

This is only available for Restore and N-Restore requests. If Nearline transfer is not available, for example no Datahub with Nearline storage is available, then a direct transfer is performed.

Default

The QOS specified in the source or destination configuration is used.

If an object to be restored has both disk and tape instances, and **Cache Only**, or **Cache and Direct** QOS is used, DIVA Core may restore the tape instance as first priority over the disk instance. This behavior depends on the *DIVAMANAGER_CACHE_QOS_USE_DISK* setting in the

Manager configuration. If set to true, DIVA Core will restore the disk instance regardless of the QOS specified.

The **Cache** transfer method is particularly important for optimum use of DIVA Core resources when the transfer speeds between tape devices and the Server vary considerably. For example, if the tape drive in the request can write data at 400 Mbps, but the source can only deliver the data at 100 Mbps, the tape drive will never achieve its optimum transfer rate. Using the **Cache** QOS, the file can be completely transferred to cache first, and the drive can complete its write operation at its maximum speed. This method enables the drive to be used for other requests in a shorter time compared to the same transfer using the **Direct** QOS.

If an object to be restored has a disk instance the **Nearline Only** or **Nearline and Direct** QOS will restore from that instance. If an object to be restored has only tape instances, the **Nearline Only** or **Nearline and Direct** QOS attempts to create a permanent disk instance and then restore from that instance. Every subsequent Nearline restore for the same object will be blocked and wait for the first restore process to create a disk instance. If the first restore fails to create a disk instance the process repeats with the next restore attempting to create a disk instance. All other restores are blocked until the disk instance has been created.

The default QOS for Restore and N-Restore requests is **Nearline and Direct**. If the restore request is a Transcode Restore, or if the destination server is a Movie2Me server, the Manager will switch the restore QOS to **Direct Only**. Other QOS types are not supported in this case.

3

Architecture Overview

The DIVA Core system is an integrated archive solution composed of several hardware and software components described in this chapter. *Refer to the Linux installation instructions in the Installation and Configuration Guide for Linux-specific directions concerning running DIVA Core components as services under Linux*.

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 will be /home/diva/DIVA/Program when running under Linux. Also, Linux paths and file names are case-sensitive.

- Hardware Components
- Software Components

Hardware Components

Multiple hardware components are required to install the software components and together comprise a complete archiving system. The following sections describe the main system components.

Storage Devices

DIVA Core performs operations among different types and formats of storage devices. Examples of usable devices include the following:

- RAID sets store data on hard disk drives
- Tape libraries automate storage on magnetic tapes. The tape library includes robotics, tape drives, and a set of tapes stored in the tape library.
- Tape drives can either be SCSI attached to the Datahubs, or through a Fiber Channel interface. When connected to a Fiber Channel Switch, they can be shared by multiple Datahubs. Sharing of resources among the Datahubs is controlled and coordinated by the Manager. The Fiber Channel Switch provides the connectivity between the Datahubs and any tape drives that are connected to it.
- DIVA Core 8.1 enables archiving operations to an Oracle Storage Cloud account, Oracle Cloud Infrastructure Storage, Amazon S3 AWS Storage account, and Sony optical libraries.

Management Stations

At least one management station is required to run the Manager software component, and the library control software supplied with the library to control the robotics. This is called the *Main Management Station*. The Main Management Station features a mirrored (*RAID1*) configuration for the data disk where the databases and all essential data are stored.

Because the Management Station is essential to the operations of the archive system, it is strongly recommended to also configure a backup Management Station. In case of failure, the main station is stopped and the backup station is started.

Datahubs and Proxyhub

Dedicated Windows or Linux servers can host the Datahub component. The Datahub software can also typically be installed directly on a production server.

DIVA Core Proxyhubs enable remote resources not visible to regular Datahubs to become visible and usable to regular Datahubs through a Proxy. A Proxyhub is simply an Datahub that acts on behalf of another Datahub. In the most common case, a Proxyhub will read or write from a remote resource at the request of a regular Datahub. The Manager tells a regular Datahub where it can find a proxy that will give it access to a needed remote resource through a new link between an Datahub and its Proxyhub. *See the Installation and Configuration Guide for Proxyhub configuration details.*

Linux-based Datahubs only support Telestream Vantage transcoding operations.

Managers

The Manager is the main component in a DIVA Core system and can be installed on Windows or Linux platforms. As a purchasable option, DIVA Core also supports Main and Backup Managers. A Backup Manager must be configured for you to use the DIVA Core Backup Service.

Network Devices

The connections between the DIVA Core system components are achieved through a 10/100BaseT or Gigabit Ethernet hub or switch.

Other Components

Other systems and components interacting with the DIVA Core system include the following:

- The applications controlling the archive operations either to move objects to the archive or to retrieve objects from the archive, and to obtain information about the archive systems or objects stored within the archive. These applications are referred to as *Archive Initiators*. Examples of an Archive Initiator are Broadcast Automation Systems, or MAM (*Media Asset Management*) applications.
- The production servers are where objects (*for example, video files*) are produced or from where they are broadcast. For example, a video server is a production server. Production servers can be the source of the objects to archive or the destination of the objects to retrieve from the archive.
- The production network is typically a high-speed LAN connecting the production servers together to allow object transfer between the servers. It also allows the connection of the Datahubs that are either attached directly to the high-speed network or through a gateway device provided by the production server manufacturer.

Software Components

DIVA Core software currently includes the following components:

- Manager
- DIVA Command (*See Chapter 1 Overview*)

- Control GUI (*Replaced with DIVA Command*)
- Backup Service
- Auto-Discovery Agent
- Checksum Support and Content Verification
- DIVAprotect (*See the DIVAprotect User's Guide for information*)
- DFM (*Drop Folder Monitor - optional*)
- SNMP Agent (*optional*)
- Customer Information Collection Tool
- VACP Converter (*optional*)
- Datahub
- Proxyhub
- Robot Manager
- DIVA Connect (*See the DIVA Connect documentation for information - optional*)
- Avid Connectivity (*See the Avid Connectivity documentation - optional*)
- Client APIs
- SPM (*See the Storage Plan Manager documentation - optional*)

Third party control software may also be provided by the library manufacturer to control the library robotics. The name of the software depends on the type and brand of the library used in the DIVA Core solution.

Email Notifications

Email notifications, the notification frequency, and the time before the first email is sent are configurable in the Configuration Utility under **DIVA Core Settings**. An email is sent to the account name if any one of the following conditions occurs:

- A configurable minimum disk space constraint is reached
- A configurable minimum number of empty tapes is reached
- A maximum number of aborted requests occurs
- The Datahub-Disk connection goes offline
- The Datahub-Drive connection goes offline
- A disk goes offline
- A drive goes offline
- A Datahub goes offline

Password Security

You cannot use the default password to log in to the Control GUI with the *administrator* or *operator* profile after DIVA Core installation is complete. You must assign an *administrator* or *operator* password in DIVA Command before you will be permitted to switch to the respective mode in the Control GUI.

If you attempt to switch to *administrator* or *operator* mode in the Control GUI without first assigning a password to the respective profile, a dialog box is displayed notifying you that you must set a password for the corresponding profile in DIVA Command. After you set the corresponding profile password in DIVA Command the first time, it no longer matters what you use for the *old password* when changing passwords.

Manager

The Manager is the main component in a DIVA Core system. All archive operations are controlled and handled by the Manager. Operation requests are sent by initiator applications through the Client API. As a purchasable option, DIVA Core also supports Main and Backup Managers.

The Manager runs as a Windows Service. You can manage the service through the Windows Services screen. The static configuration file for the Manager is manager.conf. Most settings in this file can typically be left set to the default values. Operations of the Manager can be monitored by launching the Control GUI.

You use the batch files in the Manager's bin folder to perform the following major operations:

- Start, stop, and restart the Manager Service. All of these operations can be executed using the manager batch file by specifying start, stop, or restart after the manager.bat command respectively (*for example, manager.bat start*).

  You can also terminate all running requests with a graceful_shutdown command. The graceful_shutdown command waits until all running requests have terminated before stopping the Manager instead of the abrupt shutdown that occurs with the stop command.

- Notify the Manager of any changes to the Manager's configuration using the NotifyManager batch file.
- Import tapes into a group using the importtapes batch file.
- List all active connections and end some connections (*by connection identifier*) with the ConnMgr batch file.

The manager.bat file enables you to run the Manager as a service or using a console window. Execute the batch file using the following command and parameters:

```
%DIVA_HOME%\Program\Manager\bin\manager.bat [command] [options]
```

For example:

```
%DIVA_HOME%\Program\Manager\bin\manager.bat start -conf config_file_name.conf
```

Appending the -conf (*or -f*) option after one of the following commands specifies a specific configuration file to load settings from. The manager.bat command parameters are as follows:

install (-i)
Installs the Manager as a system service.

uninstall (-u)
Removes the Manager service.

start
Starts the Manager.

stop
Stops the Manager immediately if it is running.

graceful_shutdown
Stops the Manager after all requests running at the time of the shutdown have terminated and ignores any new requests.

restart
Stops and subsequently starts the Manager.

reload
Requests that the current service reloads its settings.

status
Determines whether the service is running and displays the status.

dump
Requests that the Manager Service create a system dump.

version (-v)
Displays the Manager release information and then exits.

help (-h)
Displays help information and then exits.

Refer to the Installation and Configuration Guide for information about running Windows services in a Linux environment.

Checksum Support and Content Verification

The purpose of the Checksum Support and Content Verification program is to provide additional levels of verification for each file managed by a DIVA Core system.

During the archive process the checksum is generated automatically by the Datahub and stored in the database. This checksum is not verified until an initial read-back or restore operation is performed.

Checksum verification occurs when transferring data from a Server or when reading data from a Source Server or a storage medium. The latter occurs during the retrieval of an object from a storage medium during routine functions (***Restore**, **Copy**, **Repack**, **Transcode**, but not **Partial File Restore***), or during a read-back from storage (***Verify-Following-Write** feature*), or from the Source Server (***Verify-Following-Restore** feature*).

You view the checksum verifications and failures through the Control GUI **Manager** view, Datahub view, or other request control views. Double-clicking on the resource will display a dialog box showing verification (*or failure*) messages and the checksum information. you can pause over the checksum notation to open a smaller dialog box displaying the ***Source Server***, ***Component***, ***Checksum Type***, and ***Checksum Value***.

Note: Additional checksum verification is performed at the Oracle Storage Cloud level. *See the Oracle Storage Cloud documentation for information*.

Import Tapes Tool

The importtapes.bat batch file imports one or more tapes into a user-specified group in the DIVA Core system. You must specify the XML files created during the tape export as a command-line parameter.

This tool only imports the tape's metadata into the database and not the actual objects (*or tape*) themselves into the system. You must still insert the tapes using the **Insert Tape** functions.

See Chapter 7, "Removable Media Operations" *for more detailed information*.

Execute the importtapes.bat batch file using the following command and parameters:

```
%DIVA_HOME%\Program\Manager\bin\importtapes.bat [group_name] [mfiledir] [mfiledir]
```

The importtapes.bat command parameters are as follows:

help (-h)
Displays help information and then exits.

group_name
The tape group to which imported tapes will belong.

mfiledir
The XML files containing exported tape metadata or a folder that contains the files. The first mfiledir is required, additional entries are optional. Multiple files may be used as follows:

mfiledir1 mfiledir2 mfiledir3 mfiledir4 (*and so on*)

-skipIfNameExists

Caution: This is an advanced option and not recommended for normal use! Use of this option causes the object on the tape to become *invisible* and DIVA Core will use only the *visible* object existing in the system.

This is an advanced parameter and skips importing of objects with naming conflicts. Normally, if the object name exists the program stops and nothing is imported. This option enables skipping the existing object and continue to import the next object in the XML file.

-useImportDateAsArchiveDate
Forces use of the *Import Date* instead of the original *Archive Date* as the *Archive Date* of the object imported into DIVA Core.

Control GUI (*replaced with DIVA Command*)

Note: DIVA Command replaces the Configuration Utility starting with this release to configure a DIVA Core system. It can be run on any computer that has TCP/IP connectivity to the host running the DIVA Core database.

You use the Control GUI to monitor, control, and supervise operations in DIVA Core. Several Control GUIs can be running and connected to the same DIVA Core system at the same time.

You use the either the Windows Start menu item, or one of the following commands to launch the Control GUI in Windows:

```
%DIVA_HOME%\Program\GUI\bin\gui.bat
```

You use the following commands to launch the Control GUI in Linux:

```
cd /home/diva/DIVA/Program/GUI/bin
sh gui.sh
```

Backup Service

Caution: When using complex objects, it is *strictly required* that you use the Backup Service. Users should have an elevated awareness of error messages produced by the Backup Service.

The Backup Service ensures reliability and monitoring of both the Oracle database and metadata database backups. The Backup Service component is installed as an integral part of the standard DIVA Core system installation, and is typically installed on the same server as the Manager and Oracle database. The service enables configuration of scheduled backups through its configuration file, and manages and monitors the entire backup process.

The service generates both full and incremental database backups. Oracle database backups and metadata database backups are incrementally replicated to all remote backup systems. It is the only component backing up the metadata database and removing outdated metadata files. When you send a Delete request for a complex object and it is processed, the data is removed from the Oracle database, but the metadata database file is not deleted. It is removed by the Backup Service after the configured clean up period (*defined by the Recovery Period parameter*) has been reached.

Caution: Do not change the metadata location parameter when the system is running.

The Backup Service as a whole is comprised of two types of services, DIVA Core Backup Service (BKS) and one or more DBAgents. Both services have REST APIs such that they can be integrated with a UI component. The main backup service controls command execution, DIVA Core archives, synchronization and configuration. Each database implementation is in managed code and a minimal amount of scripting is utilized to future proof the solution. Backup configurations are also agnostic of the data contained within them such that the solution can be applied to any type of application database you would like to backup assuming the routines to do so are implemented.

Many replication locations may be configured through the BKS. These locations may be a local path or a UNC path, however the primary backup location must be local as it is used as the source of replication to all other locations. Each location may be configured with a URL to the DBAgent endpoint for that location. This is only necessary if that location is managing a remote database, in which case the database should be listed under the *Managed Databases* list. Any database in a Managed Database list will be part of the automated backup system and are eligible for restores or failovers.

One of the primary responsibilities of the BKS is to maintain a ledger of backups for each database it manages. These ledgers are located in the BKS log directory in the same folder structure the backups themselves. The default location is

C:\DIVA\Program\log\backup_service\Ledgers\<Database type>\<Database profile>\Ledger.json

These ledgers can be queried through the API and are the primary reporting structure for the active backup or restoration state of a given database. If a ledger is lost or deleted, it will be automatically created on the restart of the BKS based off of the primary backup locations contents.

Each backup is check-summed through MD5 and logged in the database ledger for each database. After a backup occurs it is replicated across all of the backup locations. After replication, if configured to do so, an archive is made using a call to the DIVA Core API to persist the backups to tape storage. The source in DIVA Core is configured in the location itself under the ***Source Name*** parameter. The name of the object will be DatabaseBackups_<Unix timestamp of the archive> and the category will be *Backups*. This only occurs after every full backup, after which a cleanup task will delete any archives that exceed the retention period.

The Backup Service periodically sends status messages to the Manager. The Manager saves all error messages received in the Manager Events Log, and also forwards messages to all connected Control GUI applications for display in a dialog box. If no Control GUIs are connected at the time of the error, no error dialog boxes will be displayed, but they are still written to the events log for later review.

The service also incorporates the ability to send emails of issues arising from the process of backing up both database files. DIVA Core must be configured to connect to an SMTP mail provider to take advantage of this feature. The email notifications are configured through DIVA Command. Identified issues are displayed on all connected GUI systems, saved to the Events log, and notification is delivered by email.

DBAgent

The DBAgent Service performs database specific tasks (*that is, backups and restores*), monitors their progress, and reports disk usage. Database maintenance functionality can easily be added if necessary, but only the specific backup tasks are currently implemented. Any number of DBAgents may be installed and configured, but only one per server/container. This is to support multi-server installations and automate access control. The DBAgent also exposes a REST API that the Backup Service will call to check the status of a backup, initiate backups/restores/failovers, and monitor disk space for configured mount points.

Configuration for the DBAgent is purposefully minimal with the only the space monitoring and backup location being required. The majority of the configuration resides in the BKS. By default, mount point configuration will monitor the backup location, the C, E, and F drives as expected by the default DIVA installation. These can be expanded to monitor other locations if necessary and can trigger alerts to DIVA when those locations are reaching their space thresholds.

A state file is created in the log directory of the DBAgent for a given database request. Backup request state files are stored in the BackupHistory directory, while respectively, Restore request state files will be in the RestoreHistory directory. These files are actively updated as the backup or restore progresses to completion and are used to gather statuses about a given action. These state files include a full log of the action itself and any files that have been created as a result of the backup process.

See Appendix , "Sample DBAgent Configuration File" *for a sample DBAgent configuration file*.

BKS Configuration

The Backup Service configuration file is monitored to allow for live updating of the configuration through the API or by direct manipulation without requiring restarting the service. By default the configuration is located here:

```
$DIVA_HOME\Program\conf\backup_service\appsettings.json
```

This path can be modified during service installation. The Backup Service contains all of the required information to connect to a database and passes that information on to the DBAgent when an action is required. The DBAgent itself also has a configuration, but it contains relatively few values.

See Appendix , "Sample BKS Configuration File" *for a sample BKS configuration file*.

Backup Initiator

A command line initiator is included in the bin installation folder. This program is a simple wrapper around the BKS API to perform backups, restores, and failovers. However it does not wait for their completion. It will offer four options when executed:

- Backup
- Restore
- Failover
- Quit

The user will select the related function they would like to perform from the additional options as follows:

Backup

1. <Database 1 -x>
2. Back
3. Quit

Restore

1. <Database 1-x>
    a. <List of restore points 1-x>
    b. Back
    c. Quit
2. Back
3. Quit

Failover

1. <Eligible failover databases 1-x>
    a. X -> Y
        i. <List of restore points 1-x>
        ii. Back
        iii. Quit
    b. Y -> X
    c. Back
    d. Quit
2. Back
3. Quit

Database Service Failover

Caution: These procedures are critical and sensitive. They should only be performed under the control of a Telestream Support Technician.

If a database or system failure occurs, where restoring from a system backup is necessary, restoration of a stored backup is accomplished using the following outlined procedures.

A failover command is very similar to the restore command though it does not guarantee the database will be up if it fails to process. During failover it is assumed that the existing data at the locations database is invalid and will be deleted prior to the failover script execution. You can failover to the same database or a different database with the same configuration.

It is recommended that failover only be used on an in-place database if the database is corrupted and in an unrecoverable state. In the case of failover to another server, the backup files from the source database are used and the existing backups for the target are essentially invalid (*although you can use them to failover to itself if necessary*). The verification of a compatible database is done at the BKS service before the command is issued to the DBAgent.

Use the following procedure to configure a standby server for failover:

1. Add the configuration in a new database profile and install a DBAgent on that standby server.
2. Add a location in the configuration that points to the main backup point for that server and add the DBAgent URL to this location configuration.

Note: Do not add the database profile to the list of managed databases unless you want active backups to be taken.

The new location will automatically be synchronized from the primary location such that all the backups are ready to be used if you need to failover.

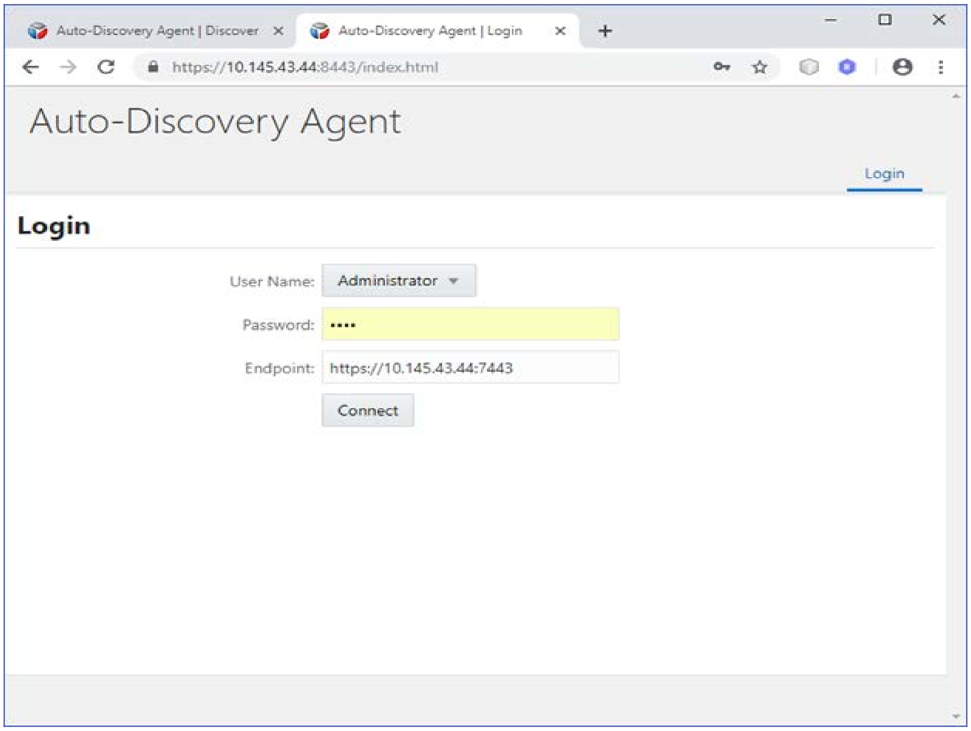
Use the following procedures to perform the failover:

1. Add the failover target to the managed list of databases for the target location.
2. Send the failover command with the source and target database profiles, along with a timestamp of the recovery.
3. Remove the source server from managed databases so it does not make active backups.

You can also recover from the loss of a Backup Service in case the server that it was running on is down by installing the backup service on another location that it was replicating to. You will need to reconfigure any existing database profiles as well as the locations associated with any prior backup locations you were replicating. This needs to be done in a stepwise fashion such that the new primary backup location can catalog all the backups into new ledgers before attempting any replication to remote locations. After this is complete, the failover procedure is the same.

Auto-Discovery Agent

The Auto-Discovery GUI is hosted on the same server as the Publisher. To access the Auto-Discovery Agent, go to the root of the Publisher end-point to load the interface.


Auto-Discovery Agent | Discover
Auto-Discovery Agent | Login
https://10.145.43.44:8443/index.html
Auto-Discovery Agent
Login

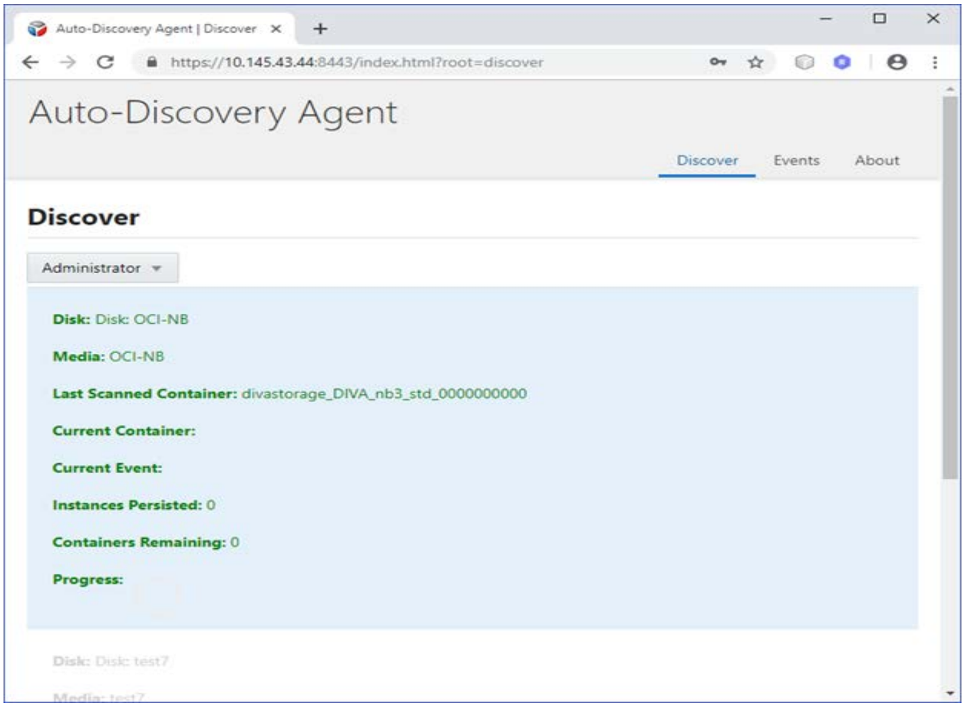
Login

User Name: Administrator

Password: ••••

Endpoint: https://10.145.43.44:7443

Login as an Administrator or User from the main screen. You must specify a valid Data Service end-point to retrieve disk data from, and the Publisher end-point. After a successful connection the main status page is shown as seen in the following figure.


Auto-Discovery Agent | Discover
https://10.145.43.44:8443/index.html?root=discover
Auto-Discovery Agent
Discover
Events
About
Administrator

Discover

Disk: Disk: OCI-NB

Media: OCI-NB

Last Scanned Container: divastorage_DIVA_nb3_std_0000000000

Current Container:

Current Event:

Instances Persisted: 0

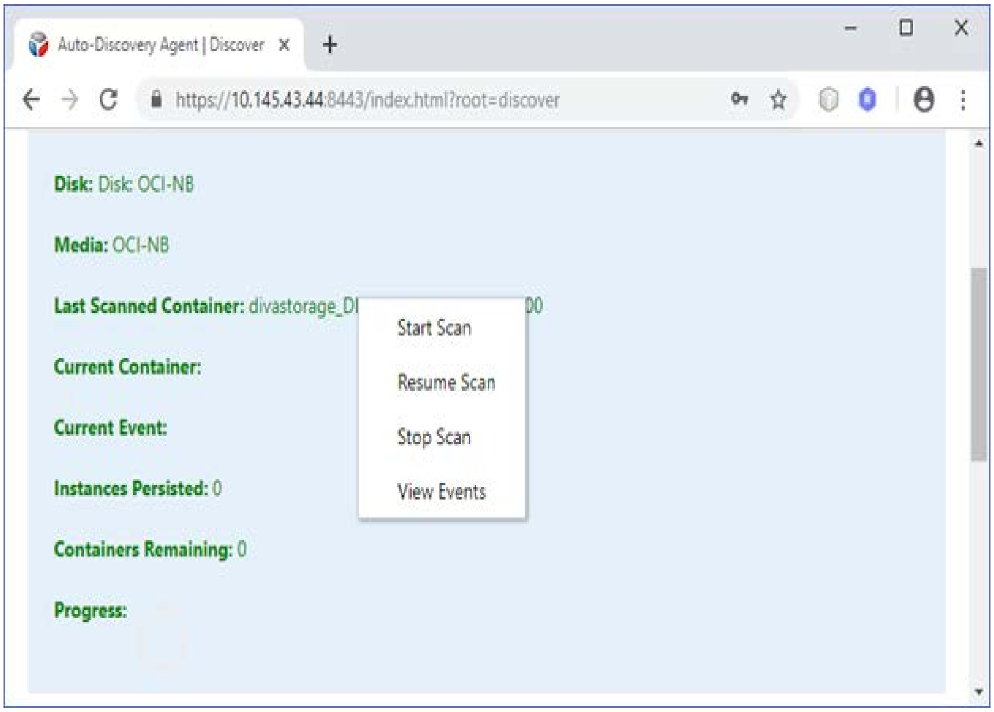
Containers Remaining: 0

Progress:

Disk: Disk: test7

Media: test7

Right-click on each disk and select one of the following operations; **Start Scan**, **Resume Scan**, **Stop Scan** or **View Events** from the pull-down menu.



Auto-Discovery Agent | Discover
https://10.145.43.44:8443/index.html?root=discover
Start Scan
Resume Scan
Stop Scan
View Events

Disk: Disk: OCI-NB

Media: OCI-NB

Last Scanned Container: divastorage_DI 00

Current Container:

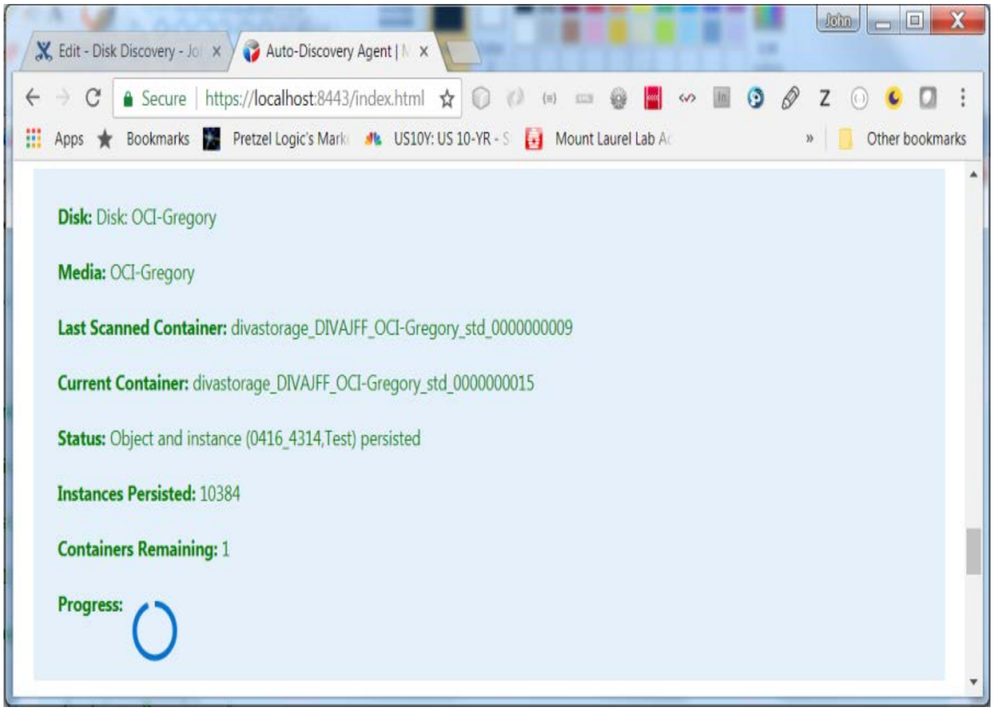
Current Event:

Instances Persisted: 0

Containers Remaining: 0

Progress:

Start Scan clears all persisted containers from the database and attempts to retrieve all content from the selected disk. Any instance that was previously persisted is skipped.



Disk: Disk: OCI-Gregory

Media: OCI-Gregory

Last Scanned Container: divastorage_DIVAJFF_OCI-Gregory_std_0000000009

Current Container: divastorage_DIVAJFF_OCI-Gregory_std_0000000015

Status: Object and instance (0416_4314,Test) persisted

Instances Persisted: 10384

Containers Remaining: 1

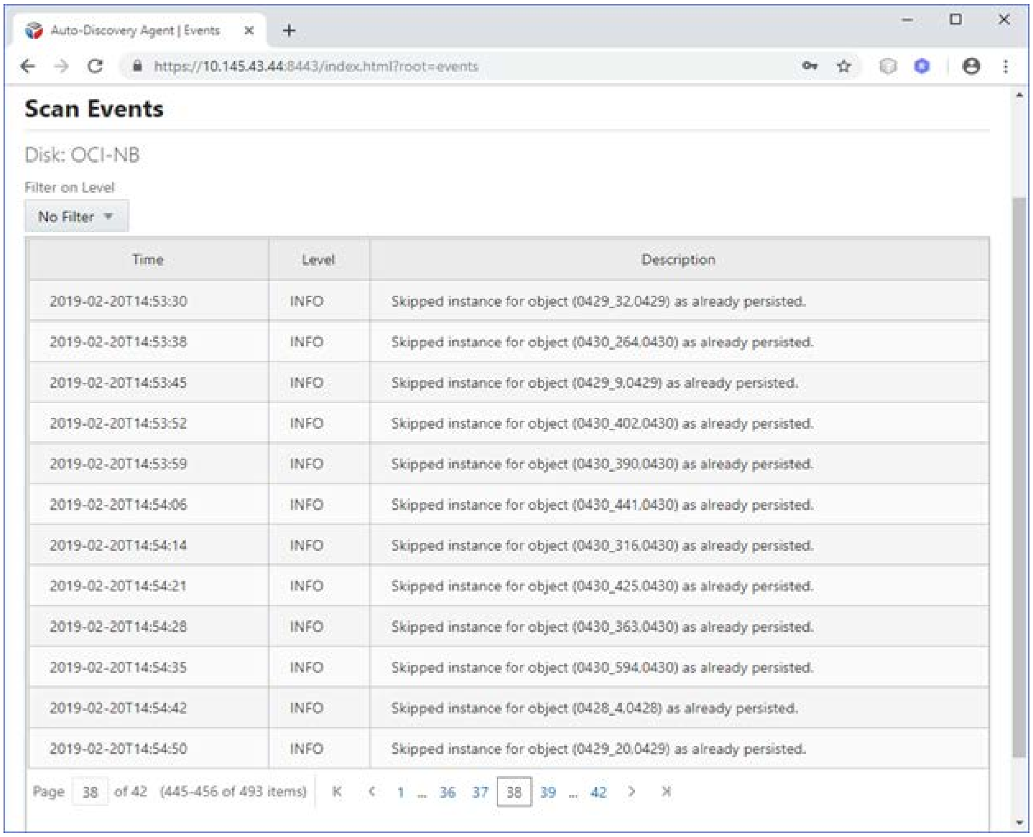
Progress:

Resume Scan attempts to resume a scan from the last successfully scanned container and instance.

Stop Scan instructs the Publisher to submit a stop request to all Datahubs currently scanning containers and terminate the disk scan. Unless a container was fully scanned, a **Resume Scan**

operation will attempt to continue a scan from the container on which a scan was terminated, starting from the discovered instance of that container with the largest UUID.

View Events displays the events screen shown in the following figure. This shows all of the events as stored in the database. The events are shown one page at a time. The page is selected at the bottom of the table.


Auto-Discovery Agent | Events
https://10.145.43.44:8443/index.html?root=events

Scan Events

Disk: OCI-NB

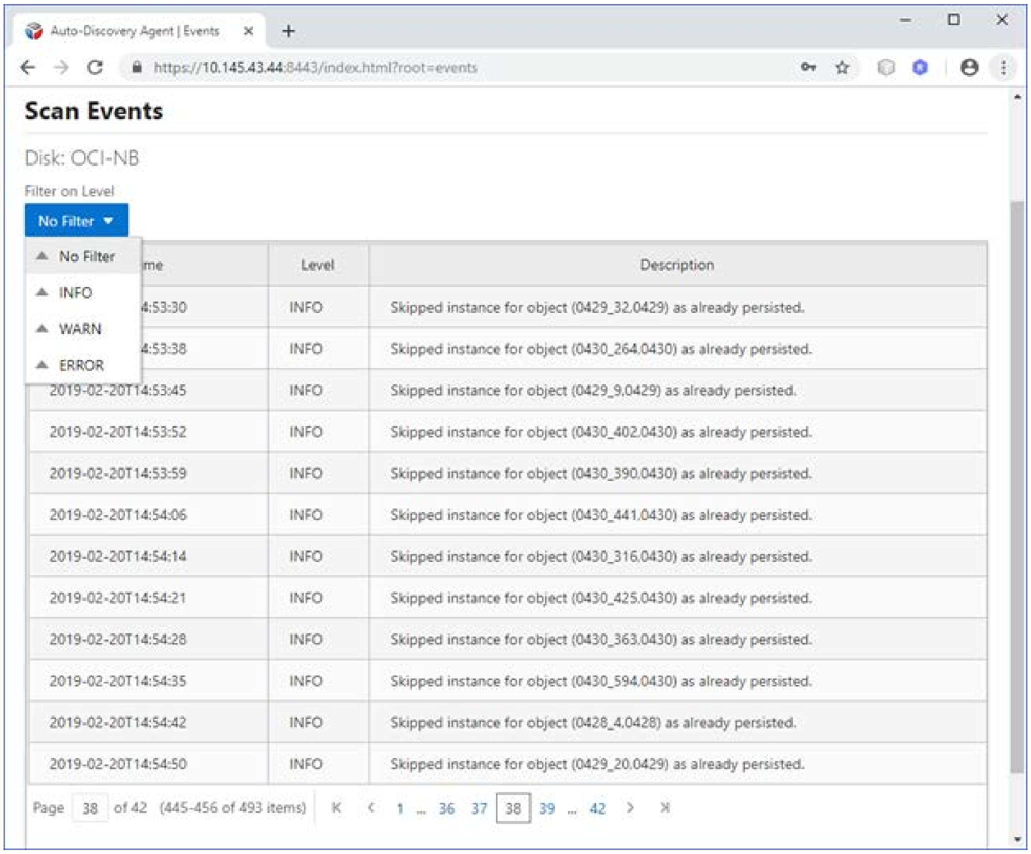
Filter on Level

No Filter

Time	Level	Description
2019-02-20T14:53:30	INFO	Skipped instance for object (0429_32.0429) as already persisted.
2019-02-20T14:53:38	INFO	Skipped instance for object (0430_264.0430) as already persisted.
2019-02-20T14:53:45	INFO	Skipped instance for object (0429_9.0429) as already persisted.
2019-02-20T14:53:52	INFO	Skipped instance for object (0430_402.0430) as already persisted.
2019-02-20T14:53:59	INFO	Skipped instance for object (0430_390.0430) as already persisted.
2019-02-20T14:54:06	INFO	Skipped instance for object (0430_441.0430) as already persisted.
2019-02-20T14:54:14	INFO	Skipped instance for object (0430_316.0430) as already persisted.
2019-02-20T14:54:21	INFO	Skipped instance for object (0430_425.0430) as already persisted.
2019-02-20T14:54:28	INFO	Skipped instance for object (0430_363.0430) as already persisted.
2019-02-20T14:54:35	INFO	Skipped instance for object (0430_594.0430) as already persisted.
2019-02-20T14:54:42	INFO	Skipped instance for object (0428_4.0428) as already persisted.
2019-02-20T14:54:50	INFO	Skipped instance for object (0429_20.0429) as already persisted.

Page 38 of 42 (445-456 of 493 items) K < 1 ... 36 37 38 39 ... 42 > ›I

You can use the provided filter is provided to show only INFO, WARN, and ERROR events as displayed in the following figure.


Auto-Discovery Agent | Events
https://10.145.43.44:8443/index.html?root=events
No Filter
INFO
WARN
ERROR

Scan Events

Disk: OCI-NB

Filter on Level

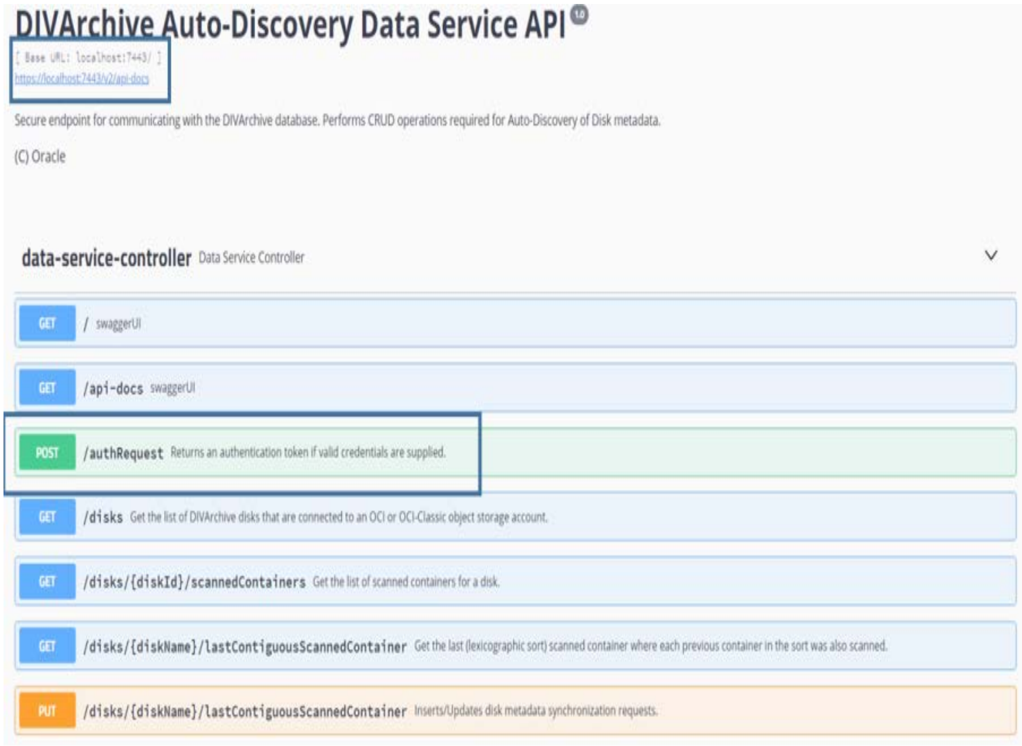
No Filter

me	Level	Description
4:53:30	INFO	Skipped instance for object (0429_32.0429) as already persisted.
4:53:38	INFO	Skipped instance for object (0430_264.0430) as already persisted.
2019-02-20T14:53:45	INFO	Skipped instance for object (0429_9.0429) as already persisted.
2019-02-20T14:53:52	INFO	Skipped instance for object (0430_402.0430) as already persisted.
2019-02-20T14:53:59	INFO	Skipped instance for object (0430_390.0430) as already persisted.
2019-02-20T14:54:06	INFO	Skipped instance for object (0430_441.0430) as already persisted.
2019-02-20T14:54:14	INFO	Skipped instance for object (0430_316.0430) as already persisted.
2019-02-20T14:54:21	INFO	Skipped instance for object (0430_425.0430) as already persisted.
2019-02-20T14:54:28	INFO	Skipped instance for object (0430_363.0430) as already persisted.
2019-02-20T14:54:35	INFO	Skipped instance for object (0430_594.0430) as already persisted.
2019-02-20T14:54:42	INFO	Skipped instance for object (0428_4.0428) as already persisted.
2019-02-20T14:54:50	INFO	Skipped instance for object (0429_20.0429) as already persisted.

Page 38 of 42 (445-456 of 493 items) 1 ... 36 37 38 39 ... 42

Documented End-points

Both the Data Service and Publisher Service have a set of documented end-points where the user can submit a request. All end-points of the Data Service except for /swagger, /api-docs, and /auth-request require an authentication token. POST a valid user name and password to the /authRequest end-point to obtain a token.



DIVArchive Auto-Discovery Data Service API 1.0

[Base URL: localhost:7443/]
https://localhost:7443/v2/api-docs

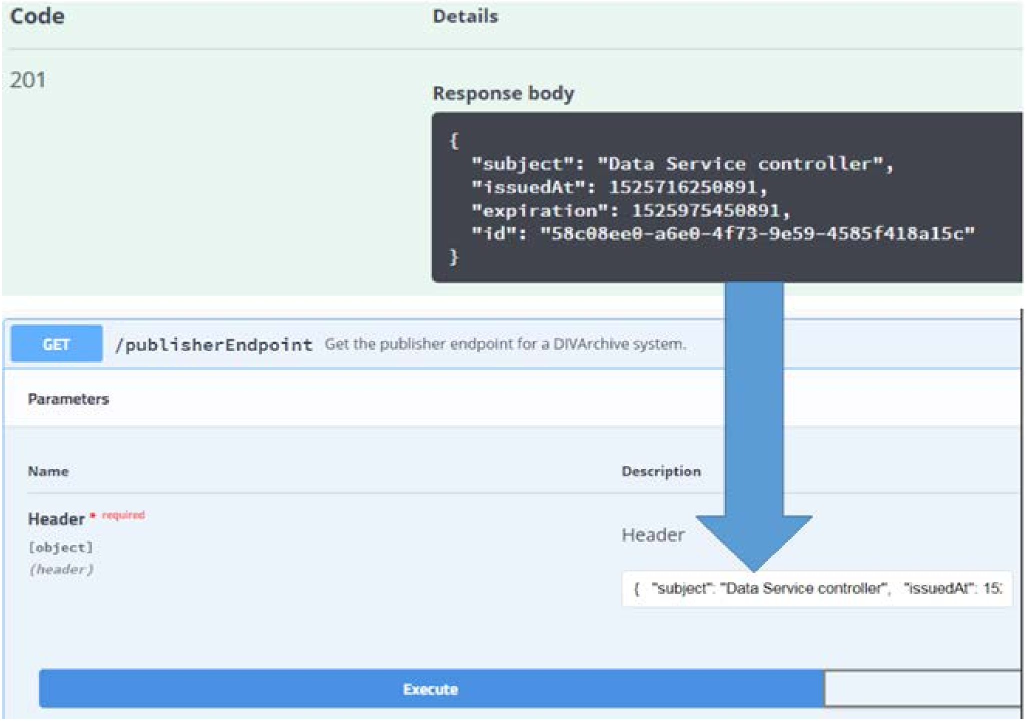
Secure endpoint for communicating with the DIVArchive database. Performs CRUD operations required for Auto-Discovery of Disk metadata.

(C) Oracle

data-service-controller Data Service Controller

GET	/	swaggerUI
GET	/api-docs	swaggerUI
POST	/authRequest	Returns an authentication token if valid credentials are supplied.
GET	/disks	Get the list of DIVArchive disks that are connected to an OCI or OCI-Classic object storage account.
GET	/disks/{diskId}/scannedContainers	Get the list of scanned containers for a disk.
GET	/disks/{diskName}/lastContiguousScannedContainer	Get the last (lexicographic sort) scanned container where each previous container in the sort was also scanned.
PUT	/disks/{diskName}/lastContiguousScannedContainer	Inserts/Updates disk metadata synchronization requests.

The user must then copy and paste the response into the header of any subsequent request.



Code	Details
201	Response body

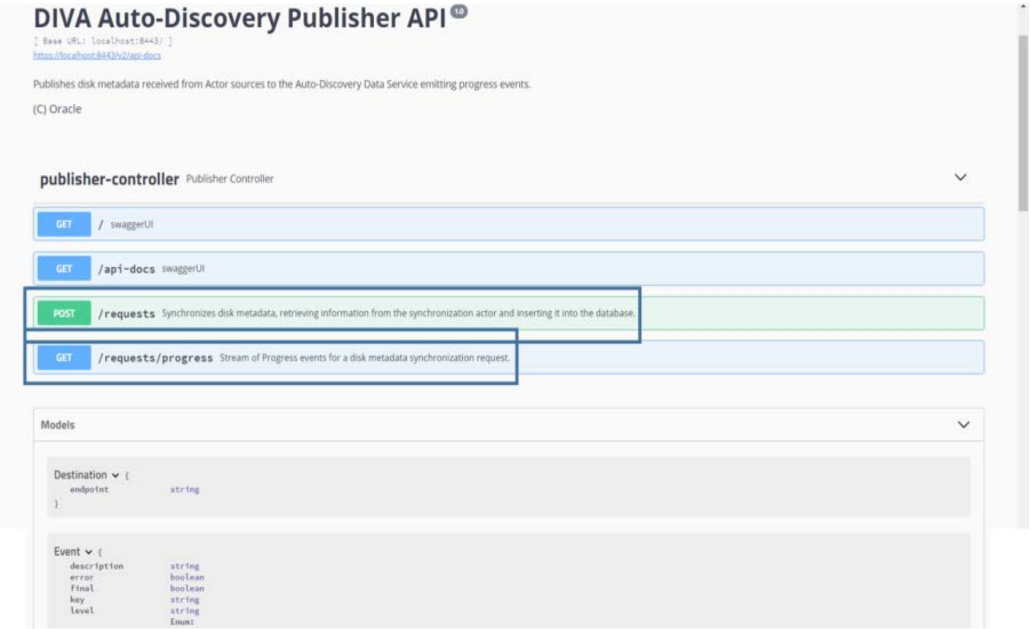
```
{
  "subject": "Data Service controller",
  "issuedAt": 1525716250891,
  "expiration": 1525975450891,
  "id": "58c08ee0-a6e0-4f73-9e59-4585f418a15c"
}
```

GET /publisherEndpoint Get the publisher endpoint for a DIVArchive system.

Parameters

Name	Description
Header * required [object] (header)	Header { "subject": "Data Service controller", "issuedAt": 15:

The Publisher end-point contains only two main end-points; one to submit a disk scan request (*/requests*) and the other to receive progress events (*/requests/progress*) as server-side events.



DIVA Auto-Discovery Publisher API 1.0

[Base URL: localhost:8443/]
https://localhost:8443/v2/api-docs

Publishes disk metadata received from Actor sources to the Auto-Discovery Data Service emitting progress events.

(C) Oracle

publisher-controller Publisher Controller

GET / swaggerUI

GET /api-docs swaggerUI

POST /requests Synchronizes disk metadata, retrieving information from the synchronization actor and inserting it into the database.

GET /requests/progress Stream of Progress events for a disk metadata synchronization request.

Models

```
Destination ⌄ {
   endpoint          string
}

Event ⌄ {
   description       string
   error             boolean
   final             boolean
   key               string
   level             string
                     Enum:
```

DFM (*Drop Folder Monitor - optional*)

DFM (*Drop Folder Monitor*) provides automatic monitoring of newly created files in up to 20 local or FTP folders (*or combinations of the two*). One (*or multiple*) files in FTP folders per object are supported. When a new file (*or FTP folder*) is identified, DFM issues an archive request automatically to DIVA Core to archive the new file or folders. After the files are successfully archived, they are automatically deleted from the Source Server.

When using DFM in a Linux environment to monitor an FTP folder, it must be configured as follows (*this is an 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

The DFM Service can be started, stopped, and restarted using the operating system Services or the DFM command line utility on each host that is running a DFM installation. When the DFM Service starts, or restarts, DFM loads and validates the configuration file. If any configuration issues are detected, the process terminates and runs diagnostics.

If the configuration validation completes successfully, DFM begins scanning all of the configured Drop Folders, checks the status of all objects that were initialized before DFM was last shutdown, and updates the internal database with the current status of the objects. After all of these checks have completed DFM is in the **Running** state.

When DFM finds files in a configured Drop Folder, it updates the internal database and requests DIVA Core to archive all files found as new objects. To avoid repeated archive requests, DFM continuously updates the archive operations status in the internal database.

If requests fail, the Status Module informs the internal database about the failure. If the number of unsuccessful request attempts reaches a preconfigured number, the object status is changed to **could not be archived** and the object is marked as incomplete. DFM logs information about the incomplete files and calls the DFM File Manager Module to move them to the Trash folder.

If the request completes successfully, the internal database is updated by the Status Module. In the case of a File Set Object, DFM removes the Metadata File and the File Set folder.

DFM terminates upon execution of the shutdown script and stops all internal processes before all archive operations are complete. After all of the modules are stopped, all internal statuses on the disk are saved in the internal database before the DFM completes shutdown.

The DFM configuration file is %DIVA_HOME%\Program\conf\dfm\dfm.conf. Service logging is performed through the log file located in the %DIVA%\Program\log\dfm\ folder. The logging configuration is in the %DIVA%\Program\conf\dfm\dfm.trace file.

The dfm.bat file enables managing DFM from a command-line interface. Execute the batch file using the following command and parameters:

```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat [command] [options]
```

Appending the -conf (*or -f*) option after one of the following commands specifies a specific configuration file to load settings from. The dfm.bat command parameters are as follows:

install (-i)
Installs the DFM module as a system service.

uninstall (-u)
Removes the DFM module service.

start
Starts the DFM module.

stop
Stops the DFM module if it is running.

restart
Stops and subsequently starts the DFM module.

status
Determines whether the service is running and displays the status.

version (-v)
Displays the release information and then exits.

help (-h)
Displays help information and then exits.

SNMP Agent (*optional*)

The SNMP (*Simple Network Management Protocol*) Agent and MIB (*Management Information Base*) supports status and activity monitoring of DIVA Core and its subsystems to a third party monitoring application through the SNMP protocol.

The SNMP agent is integrated with the Windows SNMP Service, which starts automatically when the server is started. SNMP information from DIVA Core to a monitoring application is obtained through the SNMP Agent, which in turn establishes a connection to the Manager automatically when the Manager is started.

Note: The SNMP Agent is not currently supported in the Linux environment.

You use the following procedure to configure the SNMP Service to monitor DIVA Core:

1. Install the Microsoft SNMP Services on the computer where the Manager is installed (*if not already installed*).

    1. On your server, navigate to the **Windows Key**, then **Administrative Tools**, and then **Server Manager**.
    2. Click **Manage**, and then **Add Roles and Features**.
    3. Click **Next** on each of the first four screens.
    4. Verify the *SNMP Services* are listed.

    If the SNMP Service is not listed in the services window, use the following procedure to add the service:

    1. On your server, navigate to the **Windows Key**, then **Administrative Tools**, and then **Server Manager**.
    2. Click **Manage**, and then **Add Roles and Features**.
    3. When you get to the screen showing services that can be installed, select ***SNMP Service*** and click **Add Feature**.
    4. Click **Next**, and then **Install**.
    5. After installation is complete, return to the Windows Services screen and click **Refresh** to refresh the display. The SNMP Service should now be included in the services list.

2. Stop the ***SNMP Service*** and ***SNMP Trap Service***.
3. Navigate to the SNMP installation folder.

    %DIVA_HOME%\Program\SNMP\bin

4. Confirm this folder contains the DIVAapi.dll file. If not, you can copy it from the API Visual Studio .Net Dynamic Release directory.
5. Enter the correct Manager connection information in the config.txt.ini file.

    Also, set the POLLING_RATE to 60 (*to poll for running requests every 60 seconds*), and remove the .ini from the end of the file name when saving the edited file.

6. Open the appropriate registry file and edit pathname so that it points to the SNMP path being used.

    For example, %DIVA_HOME%\Program\SNMP\bin\divasnmpagt.dll.

7. Double-click the registry file that you just edited to install the SNMP registry keys.
8. Start regedit and using the registry information in the registry file, navigate to each registry key and make sure the path in the registry is as displayed in the registry file.
9. Open the SNMP Service properties and edit the following parameters:

    - On the **Traps** tab, enter public in the ***Community Name*** field and add the IP address of the computer where traps will be viewed (*for example, the computer where the MIB browser is installed*).
    - On the **Security** tab, confirm the ***Send authentication trap*** and ***Accept SNMP packets from any host*** checkboxes are selected.
    - In the ***Accepted community names*** field, add public with **READ ONLY** rights.
    - Click **Apply**.

10. Start the SNMP service. *Do not start the SNMP Trap service.*

The SNMP Service can also be manipulated through the Windows command prompt as follows (*typically the same host as that of the Manager*):

1. Open a Windows command prompt.
2. To start the SNMP Service, enter `net start "SNMP Service"` at the command prompt. *The quotation marks are required for services with spaces in their service name*.
3. To stop the SNMP Service, enter `net stop "SNMP Service"` at the command prompt. *The quotation marks are required for services with spaces in their service name*.

Customer Information Collection Tool

The Customer Information Collection Tool is a utility feature used by Telestream Support and Development teams to collect information on the client's DIVA Core system to analyze and diagnose issues uncovered in the field. *This utility is included in the DIVA Core delivery, but is only intended to be used by Telestream personnel.*

The tool receives all customer information required for support investigations including log files, dump files, and client environment information. It receives information from all client sites in a uniform manner, and retains detailed client issue information with the originator's contact information. The tool also notifies the Telestream Development Team as soon as information is transferred to the development facility, where it is stored permanently for future issue resolution as necessary.

The CollectSysInfo.bat file enables you to collect the required information to send to the Telestream Support and Telestream Development teams for issue resolution. Execute the batch file using the following command and parameters:

```
%DIVA_HOME%\Program\Utilities\bin\CollectSysInfo.bat [parameter value]
```

Example:

```
%DIVA_HOME%\Program\Utilities\bin\CollectSysInfo.bat -EXMODULES VACP, AMCommunicator -AFTERDATE 09/25/2016
-MACHINES 172.16.3.45,172.16.3.46 -DBTYPE conf -CUST -CUSTOMER1
```

The main CollectSysInfo.bat command parameters are as follows:

-EXMODULES [MODULE_NAMES]
Excludes the specified module from collection logs and configuration files. Using -EXMODULES ALL will exclude all of the modules and only collect the DIVA Core Oracle Database dump. The default is collecting all modules.

-AFTERDATE [MM/DD/YYYY]
Collects logs only on or after the specified date. The default is collecting all available logs.

-MACHINES [IP:host_name,IP:host_name,and so on]
Collects the logs from any additional computers identified. Multiple host names are identified in a comma separated list. The default is to only collect logs for the current system where the script is running.

-DBTYPE [FULL|CONF]
Collects a full DIVA Core Oracle Database dump, or just a configuration dump. The default is collecting a full database dump.

-CUST [CUSTOMER_NAME]
The name of the customer where the logs are collected. The customer name will be truncated if it is longer than 13 characters. There is no default value for this optional parameter. If it is

not supplied as an argument the script will prompt you to enter the Customer Name during execution.

There are also several internal parameters for the script. Each of the internal parameters has a default value that can be overridden by specifying a custom value using the script options.

Example:

```
%DIVA_HOME%\Program\Utilities\bin\CollectSysInfo.bat -EXMODULES VACP, AMCommunicator -AFTERDATE 09/25/2016
-MACHINES 172.16.3.45,172.16.3.46 -DBTYPE conf -CUST CUSTOMER1 -DIVALOC C:\INSTALL\DIVA
```

The additional script parameters are as follows:

-DIVALOC
The DIVA Core installation path for all computers from where the script is collecting logs. The default value is %DIVA_HOME%.

-REMOTEDIVA
The DIVA Core installation location if additional computers are specified using the -MACHINES parameter. The path set in this parameter must be shared within the network. The default value is \\RemoteSystem\C$\DIVA.

-DUMPPATH
The location where the script will generate and output the .7z zip file. The default value is H:\.

-ORACLELOGIN
The DIVA Core Oracle Database user name and its connection details.

-CYGWIN
The Cygwin installation path. The default value is C:\cygwin\bin.

-SEVENZIP
The 7z zip tool installation path. The default value is C:\Program Files\7-Zip\7z.exe.

-TEMPDIR
The temporary directory where the script copies the logs and configuration files. This folder is created automatically at the beginning of the script execution and subsequently deleted after the script completes execution. The script will fail execution if the path set in this parameter already exists. The default value is H:/supportinfo.

VACP Converter (*optional*)

VACP (*Video Archive Command Protocol*) is a protocol developed by Harris Automation for interfacing to an archive system. DIVA Core has its own API for communicating with the 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 API commands.

The service requires a successful connection to the Manager to start. Therefore, it must be started manually either through the Windows Services component or from the command line after the Manager is running.

Use the following commands to start the VACP Service from the command prompt:

1. Open a Windows command prompt.
2. To start the VACP Service, enter net start "VACP Converter" at the command prompt. *The quotation marks are required for services with spaces in their service name*.

3. To stop the VACP Service, enter net stop "VACP Converter" at the command prompt. *The quotation marks are required for services with spaces in their service name*.

The VACPService.exe file enables you to run the VACP Converter as a service. Execute the file using the following command and parameters:

```
%DIVA_HOME%\Program\VACP\VACPService.exe command [options]
```

Appending the -conf (*or -f*) option after one of the following commands specifies a specific configuration file to load settings from. The VACPService.exe command parameters are as follows:

install (-i)
Installs the VACP module as a system service.

uninstall (-u)
Removes the VACP module service.

debug (-d)
Starts the VACP module in console mode.

version (-v)
Displays the release information and then exits.

help (-h)
Displays help information and then exits.

Datahub

The Datahub is the data mover between devices in the Network. It supports the data transfer between many different types of devices and handles transcoding operations with Telestream transcoding software (*optional*). All Datahub operations are initiated and coordinated by the Manager. You can configure one or more Datahubs to be controlled by a single Manager.

Each Datahub runs as a Windows service and will automatically start and begin accepting connections from the Manager when the Datahub host is started. Datahub services on each host may be managed from the Windows services dialog box.

When restoring the same file to the same destination twice in parallel, the behavior on Windows and Linux is different. On Windows, the first restore (*they cannot arrive exactly at the same time*) will lock the file so that the second one will terminate. On Linux, there is no such lock at the file system level. Both restores are executed at the same time, and both will write to the same file. The content of the resultant file is not predictable.

Note: Linux-based Datahubs currently only support Telestream Vantage transcoding operations.

The following list are the Datahub executable files:

%DIVA_HOME%\Program\Actor\bin\ActorService.exe command [option]
Executes commands for the Datahub Service. Appending the -conf (*or -f*) option after one of the following commands specifies a specific configuration file to load settings from. The ActorService.exe command parameters are as follows:

install (-i)
Installs the Datahub as a system service.

uninstall (-u)
Removes the Datahub service.

debug (-d)
Starts the Datahub in console mode.

version (-v)
Displays the Datahub release information and then exits.

help (-h)
Displays help information and then exits.

%DIVA_HOME%\Program\Actor\bin\scandrive.exe
Identifies the tape drives in the system. There are no command-line parameters.

%DIVA_HOME%\Program\Actor\bin\TapeReadingUtility.exe
Opens the Tape Reading Utility, which enables manually reading the tape drives in the system. There are no command-line parameters.

%DIVA_HOME%\Program\Actor\bin\VideoAnalyser.exe
Opens the Video Analyzer Utility. This utility's GUI displays the internal structure of a video format by dropping video files to the appropriate top tab of the user interface for that file type (*for example, drop a .mov file on the* ***MOV*** *tab, a .avi file on the* ***AVI*** *tab, and so on*). File information is displayed in the lower window panes. There are no command-line parameters.

Robot Manager

Although you can use DIVA Core to only manage disk storage, storage capacity can be further expanded by adding one or more tape libraries. In these cases, the Robot Manager module provides an intermediate software layer for the Manager to interact with many different types of tape libraries. It is connected to the Manager through TCP/IP.

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

The Robot Manager alerts the Manager when the collection of tapes in the associated library requires synchronization with the DIVA Core database. This functionality is specific to the SCSI Robot Manager module, and detects potential tape inventory mismatches between the DIVA Core database and the library inventory.

Potential inventory issues are trapped by the Robot Manager if the library sends unit attention codes 06h 00h 28h (*inventory may be altered*), or 06h 00h 29h (*reset occurred*). When this happens, the Robot Manager notifies the Manager so that it resynchronizes the database with the content of the library.

You access the Robot Manager Client using either the Robot Manager Client (*command line based*) or the Robot Manager Client GUI (*graphical interface*). The Robot Manager Client GUI is a graphical interface making it easy for you to interact with the Robot Manager.

Note: If intermediate robotics control software is installed between the Robot Manager and the library under control (*such as ACSLS, SDLC, or PSC*), it must be running before starting the associated Robot Manager.

When the Robot Manager Client command-line interface is started it will display a screen similar to a Windows command line. The Robot Manager Client will already be started and only the commands necessary to perform the required operations, or to display the required information, need to be entered.

The following list are the Robot Manager executable files:

Caution: Although a Robot Manager may be restarted while the Manager is running, an attempt to mount a tape to a drive while the Robot Manager is offline may cause the drives to be set *Out of Order*.

%DIVA_HOME%\Program\RobotManager\bin\RobotManager.exe command [options]
Executes commands for the Robot Manager Service. Appending the -conf (*or -f*) option after one of the following commands specifies a specific configuration file to load settings from. The RobotManager.exe command parameters are as follows:

install (-i)
Installs the Robot Manager as a system service.

uninstall (-u)
Removes the Robot Manager service.

debug (-d)
Starts the Robot Manager in console mode.

version (-v)
Displays the Robot Manager release information and then exits.

help (-h)
Displays help information and then exits.

%DIVA_HOME%\Program\RobotManager\bin\RobotManagerClient.bat [rmHost] [rmPort]
This is a command line utility to take control of the Robot Manager if the DIVA Core system is down.

rmHost
The remote host name for the connection.

rmPort
The remote host port for the connection.

%DIVA_HOME%\Program\RobotManager\bin\RobotManagerGUI.bat
This is a GUI utility to take control of the Robot Manager if the DIVA Core system is down.

Avid Connectivity (*optional*)

Avid Connectivity with DIVA Core transfers archival data to and from DIVA Core in specific video formats and enables archiving and retrieval of single clips, or a sequence of clips. Avid Connectivity is no longer packaged with DIVA Core and is a separate installation process. Additional installation is required for certain plugins for both AMC and TMC.

All operations for the AM Communicator are performed from Avid Interplay, not DIVA Core. All TM Communicator archive operations are performed from Avid, while all restore and delete operations are performed from DIVA Core.

DIVA Core 8.1 includes support for the Avid Web Services API for Archive, Restore, and Partial File Restore of clips and sequences directly from Interplay. Also included is AMC support for Interplay 3.8 and TMC support for Interplay 3.7 and 3.8.

Certain API operations used in Avid Connectivity (*such as GetByFilename and DeleteByFilename*) are not currently supported for complex objects.

See the Avid Connectivity User's Guide or contact Telestream Support for more detailed information.

Client API

The Client API is a set of documented functions enabling external applications, acting as clients, to use the services offered by the DIVA Core system.

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

The getFilesAndFolders API call is called successively to get the complete file and folder list. Normally the first time the method is called the startIndex is set to 1. For successive calls, set the startIndex to the endIndex as returned from the previous call. After all requests have been returned, a call to this method will return an empty list.

Folders do not contain a checksum, but several checksums per file are provided if available including MD5, SHA1, and so on. The returned information will identify which of the checksums is the Genuine Checksum.

See the corresponding manuals for specifications and details on the use of each API. Various APIs are available as follows:

- C++ API (*compatible with DIVA Core 6.3 and later*)
- Java API
- Enterprise Connect (*Web Services API*)
- DIVAprotectWS API

DIVAS version 2.2 includes three WS API bundles and supports both SOAP and RESTful interfaces. The WS API uses connection pooling to enhance performance, and different API bundles can be started and used at the same time.

The included API bundles are as follows:

- WS 6.5 API
- WS 7.0 API
- WS 7.1 API
- WS 7.2 API
- WS 7.3 API

Note: The new release of the Web Services API is called *DIVA Enterprise Connect*. See *the DIVA Enterprise Connect documentation library for information on installation, configuration, and operations*.

SPM (*Storage Plan Manager - optional*)

SPM (*Storage Plan Manager*) provides automatic migration and lifecycling of material within the archive based on the rules and policies defined in the SPM configuration. The SPM component is also used to trigger deletion of material from SPM managed arrays (*based on disk space watermarks*).

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. SPM will also 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.

Using FTP is currently not recommended for complex objects or Partial File Restore requests if the complex object has more than three thousand files included.

Although FTP servers are supported as Servers, the recommended practice (*currently*) is to use a local Server, CIFS, or local disk.

The current (*validated*) workaround for using complex objects with FTP servers is to slow down the FTP transfer rate of either the Server or the FTP server. However, slowing down the transfer rate decreases the performance and may also be incompatible with many workflows.

A case-sensitive FTP server is recommended for SPM metadata features.

SPM will retry failed **Copy**, **Delete** and **Restore** actions after the configured failed action retry interval as set in the SPM configuration file.

SPM supports disk cleaning based on the object's archived date. Previously, the SPM disk cleaning feature only supported cleaning based on an object's last access time and object size.

Datahubs in the Linux operating system support UNC paths for CIFS Source and Destination Servers.

The service requires a successful connection to the Manager to start. Therefore, it must be started manually either through the Windows Services component or from the command line after the Manager is running.

Use the following commands to start the SPM Service from the command prompt:

1. Open a Windows command prompt.
2. To start the SPM Service, enter net start "DIVA Core SPM" at the command prompt. *The quotation marks are required for services with spaces in their service name*.
3. To stop the SPM Service, enter net stop "DIVA Core SPM" at the command prompt. *The quotation marks are required for services with spaces in their service name*.

The SPMService.exe file enables managing SPM from the command-line interface. Execute the file using the following command and parameters:

```
%DIVA_HOME%\Program\SPM\bin\SPMService.exe command [options]
```

Appending the -conf (*or -f*) option after one of the following commands specifies a specific configuration file to load settings from. The SPMService.exe command parameters are as follows:

install (-i)
Installs the SPM module as a system service.

uninstall (-u)
Removes the SPM module service.

debug (-d)
Starts the SPM module in console mode.

version (-v)
Displays the release information and then exits.

help (-h)
Displays help information and then exits.

See the SPM (Storage Plan Manager) Guide for detailed information.

Miscellaneous Utilities

DIVA Core includes various miscellaneous utilities, some of which are associated with the modules previously listed in this chapter. The included utilities are as follows:

%DIVA_HOME%\Program\Utilities\bin\DIVAConfigurationPrinter.bat
Prints the current DIVA Core configuration. There are no command-line parameters.

%DIVA_HOME%\Program\Utilities\bin\DivaScript.exe
This utility enables using command line orders to execute requests and operations.

%DIVA_HOME%\Program\Utilities\bin\GetVersion.exe [application_path]
Returns the release number for a specific application. The application_path is the valid path to the application being checked.

%DIVA_HOME%\Program\Utilities\bin\rdtu.bat
The RDTU (*Recover Damaged Tape Utility*) recovers object instances that reside 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 array*) within a local or remote DIVA Core system. There are no command line parameters. The settings and configurations are defined in the rdtu-conf.xml configuration file.

4

Starting and Stopping DIVA Core

This chapter describes the steps required to start and stop DIVA Core. See the Linux installation instructions in the *Installation and Configuration Guide* for Linux-specific directions associated with running DIVA Core components as services in a Linux environment.

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 is /home/diva/DIVA/Program when running under Linux. *Linux commands, paths and file names are case-sensitive.*

- Starting DIVA Core
- Stopping DIVA Core
- Manager Failover Procedures

Starting DIVA Core

To start the DIVA Core system you must first start the hardware, and then the software in the sequence as described in the following sections.

Starting DIVA Core Hardware

Perform the following steps in sequence to start all of the DIVA Core hardware components. Wait for initialization of each hardware component to complete before moving to the following step.

1. Confirm that all required devices are installed. If they are not installed, you must install them before proceeding any further.
    1. Libraries and Drives
    2. SAN RAID Arrays
    3. Fiber Channel Switches
    4. Networking Devices
    5. Terminal Concentrator
    6. Graphical Front End Hosts
    7. Library Manager Host
    8. External Direct Attached Devices
    9. Manager Hosts
    10. Datahub Hosts

2. Power on the Libraries and Drives.
3. Power on the SAN RAID Arrays.
4. Power on the Fiber Channel Switches (*if installed*).
5. Power on the Networking Devices.
6. Power on the Terminal Concentrator (*if installed*).
7. Power on the Graphical Front End Hosts (*if installed*).
8. Power on the Library Manager Host (*if installed*).
9. Power on External Direct Attached Devices.
10. Power on Manager Hosts.

    In installations where two Manager Hosts are installed, it may be required to always start the Main manager first, and then the Alternate (*or Backup*) Manager at a later time. Consult with your Telestream Installer to determine if this is applicable to your installation.

11. Power on the Datahub Hosts.

Hardware start is complete if everything powered on successfully.

Starting DIVA Core Software

The following steps describe the required order that the software components of a DIVA Core system must be launched. Some software components may be set to launch automatically when the host is started (*for example, the Datahub Service*).

Note: Some DIVA Core Windows Services may be disabled, or set to be launched in manual mode, due to the configuration and settings done by Professional Services during the installation.

The management of each DIVA Core software component, whether manually or automatically initiated, is covered in Software Components. Perform the following steps in sequence to start all of the DIVA Core software components:

1. Confirm that all required components are installed. If they are not installed, you must install them before proceeding any further.
    1. Library Manager
    2. Library
    3. Manager
    4. Backup Service
    5. Complex Objects (*if in use*)
    6. DIVA Connect
    7. VACP Converter
    8. SPM (*Storage Plan Manager*)
    9. DFM (*Drop Folder Monitor*)
2. Launch the Library Control software.
    1. ACSLS

    2. PCS
    3. SDLC

3. Launch the Robot Managers.
4. Launch the Datahubs.
5. Launch the Manager.
6. Launch DIVA Connect.
7. Launch the VACP Converter.
8. Launch the SPM (*Storage Plan Manager*).
9. Launch the DFM (*Drop Folder Monitor*).

Software start is complete if everything powered on successfully.

Stopping DIVA Core

You stop DIVA Core in the reverse order from starting the system. You shut down the software first and then the hardware. The following sections describe the required procedure to fully shut down DIVA Core.

Shutting Down the Software

To ensure that requests currently still in progress are not prematurely aborted by shutting down the DIVA Core system, it is recommended the Manager be stopped first, because any requests currently active will be completed before the Manager will complete the shut down.

See the Manager section for Manager shut down procedures. When the Manager is shut down all archive operations are ceased. It is not necessary to stop other software components before shutting down the host computer where they are installed.

Shutting Down the Hardware

Use the following procedure (*in sequence*) to shut down all DIVA Core related equipment and devices:

1. Shut down the Manager Host.
2. Shut down the Datahub Hosts.
3. Power off all External Direct Attached Devices.
4. Power off Graphical Front End Hosts (*if installed*).
5. Power off Terminal Concentrator (*if installed*).
6. Shut down the Library Manager Host (*if installed*)
7. Power off Network Devices.
8. Power off Fiber Channel Switches (*if installed*).
9. Power off SAN RAID Arrays (*if installed*).
10. Power off Library and Drives.

Hardware shut down is complete if everything powered off successfully.

Manager Failover Procedures

Caution: These procedures are critical and sensitive. They should only be performed under the control of a Telestream Support Technician.

The following steps are required to failover a DIVA Core Manager to the Backup when the database is still accessible on the original Manager:

1. Ensure all contents of the DIVA folder from main Manager exist in the Backup Manager (*particularly the correct .conf files*). If the do not exist move the .conf files to the Backup Manager.

Caution: Make sure to confirm the Backup Manager has the correct DIVA binary files including major/minor version, patches, and proper database version. Always keep a backup of the original DIVA folders if making any file changes.

2. Confirm all services are installed, for example DFM, Manager, Backups, SPM, Oracle, and so on, on the Backup Manager machine. If not, the services must be installed before proceeding. Ensure the services are at the same version and patch level as the main Manager.
3. Stop all services and export the database from the original Manager. *Contact Telestream Support if the database is not accessible due to failure.*
4. Create a new DIVA user on the Backup Manager using the -notable option, then import the database to the Backup Manager and verify the count of archived objects is correct from the Original Manager to the Backup Manager. This can be done with the following query in SQL;

   ```
   SELECT COUNT(*) AO_OBJECT_NAME from DP_ARCHIVED_OBJECTS;
   ```
   Contact Telestream Support if you need assistance exporting and importing the database.
5. Change the Backup Manager IP to the Original Manager IP by first applying a placeholder IP on the Original Manager.
6. Confirm the configuration is valid in the manager.conf, robotmanager.conf, spm.conf, and all disk and file paths in the configuration are accessible from the Backup Manager machine.
7. Enable and start all services and confirm Backup Manager is running as anticipated; monitor activities.

Cluster Failovers

Use the following procedures if a cluster fails to initiate:

1. Check that the backups are synced on the Active Node and Backup Manager.
2. Stop all DIVA Services from the Microsoft Cluster on the Active Node.
3. On the Active Node, run SELECT COUNT(*) AO_OBJECT_NAME from DP_ARCHIVED_OBJECTS;.
4. Create an export of the current database from the Active Node.
5. Stop the DB Services on the Active Node from the Microsoft Cluster Manager.
6. Start the DB Services on the Backup Manager server.
7. Recover the database from the backups. *Contact Telestream Support if you require assistance recovering the database*.

8. Start the DIVA Services on the Backup Manager and run some tests to confirm functionality.
9. When all testing is successful, stop all Backup Manager services, and restart all services from the Microsoft Cluster Manager. Verify all operations are functioning normally.

5

DIVA Command Operations

Caution: DIVA Command is only intended for experienced users. Incorrect or incomplete changes in DIVA Command can adversely affect DIVA Core operations, possibly delete data from the archive, or prevent the Manager from running. If you are unsure about making changes contact Telestream Support for assistance.

The new DIVA Command web-based interface has replaced the DIVA Core Configuration Utility starting with DIVA Core release 8.1.

All information relating to objects that have been archived including (*but not limited to*) where they are stored, tape locations, DIVA Core Configuration, and so on, is stored within the DIVA Core database.

You access the DIVA Core database through DIVA Command. Although the utility is only intended for use by trained administrators, from an operational perspective some functions of the utility may need to be occasionally accessed by non-administrators.

Operators should only access DIVA Command if they need to perform alterations of the attributes for one or more tapes, such as repack status or Set ID.

The utility can be installed and run on any host with TCP/IP connectivity to the DIVA Core database, Manager, and Robot Managers.

Disks or tape drives that have been set to **Out of Order** (*displayed in the **Disks** or **Drives** views in the Control GUI*) must not have the status altered until the source of the problem is investigated and rectified by an administrator.

- Launching DIVA Command
- DIVA Command Screens

Launching DIVA Command

You launch the DIVA Command in your web browser. Use the following steps to connect to the DIVA Core database:

1. Open your web browser.
2. Navigate to http://172.16.10.18:8765/.

If the connection was successful, you will be able to view all of the items in the DIVA Core area of the screen. If the connection fails, an error message will be generated in the notification area including the error code returned from Oracle. If you cannot connect, contact Telestream Support.

DIVA Command Screens

The screens in DIVA Command are different than what user would see in the Configuration Utility. *See the Installation and Configuration Guide for descriptions of the screens and applicable configuration procedures. Contact Telestream Support for more information on each tab*.

6

Control GUI Operations

The Control GUI is a software utility that connects to both the Manager, and the DIVA Core database, to monitor, control, and supervise operations in DIVA Core. Multiple Control GUI instances can be operated simultaneously from any computer that has TCP/IP connectivity to both the Manager and DIVA Core database. The Control GUI is based on Oracle Java and is platform independent.

The Control GUI is not intended for the intensive archive operations of a DIVA Core system. Typically, archive operations are initiated to DIVA Core from a Broadcast Automation or Media Asset Management system. The Control GUI is intended to supplement these operations rather than replace them.

The Control GUI provides the following features:

- Monitoring of the requests that have been submitted either through the Client API, or from a Control GUI.
- Monitoring of the status of the Datahubs, Drives, and Disks connected to DIVA Core.
- Initiate and submit all Client API available commands, such as **Archive**, **Restore**, **Partial File Restore**, and so on, to DIVA Core for execution.
- Management of tapes for each library controlled by DIVA Core (*such as internalizing, externalizing, and tape repacking*).
- Interrogation and data mining of the DIVA Core database.

See the section related to Requests *or contact Telestream Support for more information*.

This chapter includes the following information:

Launching the Control GUI and Connecting to the Manager

Use the following procedure to launch the Control GUI and connect to the Manager:

1. Double-click the Control GUI icon (*typically on the computer desktop*) to start the Control GUI.

2. After the interface loads, click the **Connect** icon on the top left of the screen. The **Connect** icon is the first icon on the left, above the **Home** tab.
3. In the Connect dialog box, enter the Manager's IP address and TCP Port in the ***IP address*** and ***Port*** fields.
4. Click the **Connect** button.

If the connection is successful, the Control GUI will show *Connected* in the connection status area on the left at the bottom of the screen. If the Control GUI cannot establish a connection the Manager, it will attempt to connect to only the DIVA Core database. Consult your Site Configuration for the connection parameters specific to your site.

User Permissions

After the connection to the Manager is established, the Control GUI will only allow you to monitor operations, and retrieve data from the database. This is known as the *User* profile.

Not all command functions are accessible while in the *User* profile mode. This permits situations where monitoring is required but no commands are permitted to be sent to DIVA Core.

To issue requests to DIVA Core, such as Archive or Restore requests, or to eject a tape from a library, you must change to the *Administrator* profile.

The *Administrator* profile is password protected. The password for this profile is set during system installation (*contact Telestream Support for help with this password*).

An *Operator* and *Advanced Operator* profile are also present in the Control GUI profiles. The difference between the two profiles is **Insert** and **Eject** tape commands are included in the *Advanced Operator* profile. During normal operation, you use the *Operator* profile unless you are inserting or ejecting a tape. When the profile is *Administrator* or *Advanced Operator*, you must select a tape or tapes before all available commands become active.

Use the following procedure to change profiles:

1. Click the **Profile** icon on the top left of the screen. The **Profile** icon is the third from the left and has two silhouettes on it.

   You can also click the orb on the top left of the screen and click the **Change Profile** menu item in the resulting menu.
2. Select the desired profile from the **Select New Profile** list.
3. Enter the selected profile's password in the **Password** field.
4. Click **OK** to load the new profile.

Control GUI Preferences

You access the Control GUI Preferences from the **Start** orb on the top left of the screen as follows:

1. Click the **Start** orb on the top left of the main screen.
2. Click the **Tools** menu item.
3. Click the **Preferences** menu item to display the Preferences dialog box.
4. On the **Preferences** tab, you set the ***Current Requests*** in the Current Requests field.

   This option identifies the maximum number of requests (*including completed or aborted requests*) displayed simultaneously in the ***Manager Current Requests*** view. When the number of requests displayed exceeds this number, the oldest request is removed as each new request is added.

5. On the **Preferences** tab, you set the ***Max Rows Requestable for Database*** in the ***Max Rows Requestable from Database*** for each of the fields. When a database query is executed, the maximum number of lines returned is limited to these values. If the results of a query exceed this number, the number of queries shown will be the designated number (*maximum*), and a window will be displayed stating *There are nnnn rows matching filters. Change filters to reduce this number under nnnn*.

   - ***Tapes***
   - ***Archive Objects***
   - ***Require/Release***
   - ***Logged Events***
   - ***Logged Requests***
   - ***SPM Actions***

You set the visual window type and color theme on the **Look-and-Feel** tab using the ***Look-and-Feel*** list and the ***Theme*** list.

On the **Fonts** tab, you can leave the fonts at the system default (*select the **use system default fonts** option*), or customize the display fonts used in the Control GUI. The default font for the Control GUI is *Arial Unicode MS* - which supports Unicode characters. Use the following procedure to select custom fonts:

1. Click the **Fonts** tab.
2. Select the ***use custom fonts*** option.
3. To the right of the ***Labels*** field, click the **Select** button to select the font for ***Labels***.
4. To the right of the ***Fields*** field, click the **Select** button to select the font for ***Fields***.

Manager Log Level Configuration

The Manager's log level is configurable through the Control GUI. If a greater level of detail is required to examine the Manager's activity, you can change the log level without restarting the Manager. Use the following procedure to change the log levels:

1. Click the **Start** orb on the top left of the main screen.
2. Click the **Tools** menu item.
3. Click **Modify Log Levels** menu item.

   The Modify Manager Log Levels dialog box is displayed.
4. Use the ***Trace Level*** list to select the logging level for trace logs.
5. Use the ***Service Level*** list to select the logging level for service logs.
6. Click **OK** to save your changes.

Control GUI Dashboard and Quick Launch Buttons

The Control GUI look and feel resembles other Windows-based applications. The Control GUI Dashboard displays information at a glance as soon as the application is started.

The dashboard presents statistics representing system data in the form of bar graphs. Daily and lifetime statistics are displayed below the graphs.

The following graphs are displayed by default:

Daily Operations

This graph reflects a seven day window of **Archive, Restore**, **Partial File Restore**, **Delete**, and **Copy** operations performed, and a thirty day average of these operations.

Daily Data Transfers

This graph reflects a seven day window of outgoing and incoming data movement, and a thirty day average is displayed.

Storage Distribution

This graph reflects data storage distribution to Nearline disk arrays, tapes in an online ACS, tapes in an offline ACS, and tapes on the shelf.

Monthly Storage Trend

This graph reflects the monthly data archived and deleted. The twelve month average reflects data movement over the past twelve months. Activity in the current month is used for the twelve month average calculation.

Tapes Status

This graph reflects the total number of online tapes managed by DIVA Core that are empty, partially used, and totally full.

Resource Utilization

This graph reflects resource utilization of Datahubs, tape drives, and transcoders.

Quick Launch Buttons

The following quick launch buttons are located at the top of the Control GUI interface:

Connect to Manager

This is the first button on the left and enables connecting to the Manager. The button has two computers and plus sign images.

Disconnect from Manager

This is the second button from the left and disconnects the Manager connection. This button has two computers and minus sign images.

Change User Profile

This is the third button from the left and enables access to different user profiles (*User*, *Administrator*, *Operator*, *Advanced Operator*). This button has two silhouettes images.

Display the Dashboard

This is the fourth button from the left and displays the dashboard screen when clicked. This button has a clock image and bar graph image.

Display Manager

This is the fifth button from the left and displays the Manager screen when clicked. This button has a computer image and two arrows, one right-facing and on left-facing.

Display Archived Objects

This is the sixth button from the left and displays the Archive Objects screen when clicked. This button has a video play button image.

Display Logged Requests

This is the seventh button from the left and displays the logged requests screen when clicked. This button has a log file image.

Display Logged Events

This is the last button on the right and displays the logged events screen when clicked. This button has a calendar image.

Control GUI Toolbars and Navigation

DIVA Core's toolbars and navigation use a Windows-style ribbon bar and tabs. You can also perform various functions from the **Start** orb on the top left of the screen.

The **Start** orb contains the following menu items:

- **Connect** (*to Manager*)
- **Disconnect** (*from Manager*)
- **Change Profile**
- **Connection Information**
- **Tools**
  - **Export Current View**
  - **Print Current View**
  - **Generate Thread Dump**
  - **Modify Log Levels**
  - **Preferences**

You can retrieve the release information of your Control GUI by clicking **About DIVA Core CSM** on the DIVA Core **Start** orb.

You can retrieve the connection information of your Control GUI by clicking **Connection Information** on the DIVA Core **Start** orb.

The following sections describe the tabs on the ribbon bar. Each tab contains icons to display different screens as described.

Home Tab: Dashboard

Clicking this button will take you directly to the DIVA Core main dashboard screen that displays general system information and statistics. The dashboard view is described n the previous section.

Home Tab: Manager (*Current Requests View*)

The Current Requests view primarily displays the current requests submitted to DIVA Core that are currently in, or pending, execution. Requests that have completed, aborted, or encountered warnings during execution are also displayed. This feature only applies when the Control GUI is connected to DIVA Core. Completed or aborted requests before the connection are not displayed. The number of pending, executing, completed, and aborted requests displayed while connected depends on the Control GUI preference settings. Right-clicking a request produces a context menu with additional options.

Request Steps

The ***Step*** column indicates the current operation of the request being performed by DIVA Core. A short description of each step is as follows:

Mounting

A tape is being inserted into a drive. The mounting step is completed once the tape is fully threaded, positioned at the tape header, and the DIVA Core label on tape verified with that of the barcode label.

If the label does not match, the request will be aborted, and the tape will be set to **Not Writable**. This situation could occur if there is a mismatch in the Datahub-Drives configuration, or the tape has already been used in another archive system (*and therefore has a tape label not usable by DIVA Core*). The latter example is a protection feature in shared library environments, where the tape has not been set to ***Set ID*** 99 (*that is, not in use by DIVA Core*).

Tapes from other archive systems must first have their tape label cleared before using them in DIVA Core. Contact Telestream Support for assistance to use such tapes.

Dismounting

This step involves ejecting a tape from a drive. A Datahub first issues an **Eject** command to the drive (*in which it rewinds, unthreads and ejects the tape*), and the Robot Manager issues a dismount command to the library to return it to a tape bin. If the drive cannot complete the request, the request is terminated and the drive set to **Out of Order**.

Positioning

When reading from tape, the tape is positioned to the selected object. When writing to tape, the tape is positioned to the End of Data (*that is, to the position where the last object was written*). If this process takes too long, DIVA Core will time out the operation and attempt to dismount the tape. If this also fails, the drive will be set to **Out of Order**.

Reading

The object displayed in the ***Object*** column is being read from tape. If this step takes too long (*for example, the drive is in a hung state*), DIVA Core will time out the step and attempt to use another drive (*or instance if available*).

Writing

The object displayed in the ***Object*** column is being written to tape. If this step takes too long, (*for example, the drive is in a hung state*), DIVA Core will attempt to write the object to another tape in another drive.

Deleting

This step is rewriting the tape's label and moving the End of Data pointer therein before writing to the tape in an archive operation. This will be seen on a tape which has previously been used by DIVA Core but has since had all objects deleted and consequently returned to the Unused Tapes Sets.

Transferring

Data is either being transferred to a Source Server from an Datahub cache or from a destination to the Datahub cache.

Waiting for Operator

This step holds the request in a suspended state and is waiting for human intervention, such as inserting tapes in library's Cartridge Access Port.

Waiting for Resources

DIVA Core resources required for this request are currently in use by another request and the request will be executed when they become available. Resource availability can also be influenced by the ***Request Priority*** of other requests much lower in the queue.

DIVA Core incorporates an intelligent feature (*which must be enabled in the DIVA Core configuration*) whereby requests with a lower request priority will be boosted over higher priority requests if they involve a tape that is already mounted from a previous request. This feature can substantially reduce the amount of tape mounting and dismounting, and speed up the executions of all requests overall.

Clearing Completed Requests

Completed, **Aborted**, **Partially Aborted**, or **Cancelled** requests can be cleared from the Current Requests Queue by clicking the **Clear** or **Clear All** buttons on the **View** tab or the **Current Requests** context menus.

Canceling a Request

You can cancel a running or pending request by first selecting the specific request to be canceled, and either clicking **Cancel** from the command menu or from the Current Request context menu.

Note: The current operation (*or step*) on a request that is currently being executed may need to complete before the request is actually canceled by DIVA Core.

Changing the Request Priority

If there are several pending requests in the Current Requests Queue, DIVA Core will process each request based on its ***Request Priority***.

If you need a specific request to execute before (*or after*) the requests that precede it in the queue, you can manually adjust that request's priority to be higher (*or lower*) than that of the preceding requests. Raising (*or lowering*) a request's priority can also be achieved through the Client API using a third party archive initiator.

Raising a pending request's priority does not stop, or pause, any requests that are currently executing. This simply changes the order in which the pending request will be processed by DIVA Core, except if a resource being used by a running request (*such as a specific tape*) becomes available after that running request has completed an operation. The order of request execution may also be influenced if the DIVA Core_PRIORITY_TIER setting in the DIVA Core configuration is enabled (*that is, a request lower in the queue will have this value added to its request priority if it involves a tape already mounted*).

By default, DIVA Core will periodically increment the request priority of all requests in the queue. This prevents low priority requests (*such as **Copy to Group***) from continually being overridden by higher priority requests and getting stuck indefinitely in the queue.

Retrying a Request

You can resubmit a previously completed or failed request to DIVA Core using the **Retry** command. This is useful for resubmitting similar requests where few of the details change between them. For a request that was terminated (*for example, because a parameter was incorrectly entered, or a Server was briefly offline*), the failed request can be retried without having to submit a completely new request.

Home Tab: Datahubs

The Datahubs view provides an indication of the status of each Datahub defined in DIVA Command and any currently running requests. This view is displayed by selecting **Datahubs** in the **Home** tab.

Selecting one of the Datahubs will display the currently running request on that Datahub in the window below it. If the Manager cannot establish a connection to an Datahub, it is displayed as **Off**. Right-clicking on an Datahub displays that Datahub's configuration.

Home Tab: Robot Managers

The **Robot Managers** button on the **Home** tab displays the robots identified in the DIVA Core system.

Home Tab: Libraries

The Libraries view gives the information and an indication of the status of each of the libraries connected to DIVA Core. This screen displays the ***Serial Number***, ***Name***, ***Type***, ***ACS***, ***Status***, ***First Utilization Date***, ***Total Tapes***, ***Total Data Stored***, ***Total Capacity***, and ***Free Capacity*** for the connected libraries.

You double-click the library to display the Library Entry Detail screen. This view offers information concerning the specified library.

Home Tab: Drives

The Drives view displays the status of each tape drive in the libraries connected to DIVA Core, what (*if any*) tape is mounted in each drive, and current operations being performed on the tape in the drive. Online or offline status for a drive is configured in the **Drives** tab of the **DIVA Configuration, Media Storage** menu item in DIVA Command.

If DIVA Core encounters a problem with a particular drive, it will set the drive to **Out of Order** as a safety measure. When a drive is set to this state it will not be used for DIVA Core operations.

Note: If a drive has been set to **Out of Order**, the cause of the error must be investigated before setting the drive back to **Working Well** in DIVA Command.

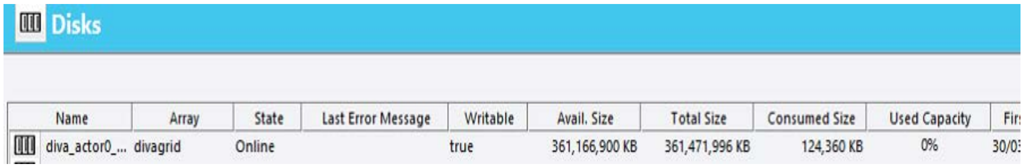
Home Tab: Disks

The Disks view displays the online status and capacities of disks configured in DIVA Core. The status of a disk can be set through the **Disks** tab of DIVA Command. If DIVA Core has automatically set a disk to **Out of Order**, the cause of the error must be investigated before setting the disk back to **Working Well**. If DIVA Core encounters an I/O error with the disk, it is set to **Out of Order** automatically by DIVA Core.

The column titled ***Consumed Size*** represents the space in kilobytes consumed by the content on disk. This column is especially useful for cloud accounts with unlimited disk space, because it provides visibility into the amount of content stored in the cloud.

Note: A disk linked to an OCI storage account will always report a Consumed Size of 0.

The ***Cloud Storage Class*** column associated with the array is also displayed in this view. Non-cloud disks have a storage class of **NONE**. OPC Cloud disks have a storage class of **Standard** (*immediately available for download from the cloud*) or **Archive** (*requires a maximum four hours to download from the cloud*). OCI cloud disks only support STANDARD storage. The following figure displays the configured values of an Amazon S3 disk in the Disks tab:



Disks

Name	Array	State	Last Error Message	Writable	Avail. Size	Total Size	Consumed Size	Used Capacity	Fir:
diva_actor0_...	divagrid	Online		true	361,166,900 KB	361,471,996 KB	124,360 KB	0%	30/0:

Viewing Storage Options

You can view the disk storage options on the **Home**, **Disks** screen in the Control GUI. When you click an object in the **Manage**, **Objects** screen, the *Object Properties* window is displayed. You can see the instances storage options for the selected object in the *Instances* area of the window.

Home Tab: Tapes

You display this view by selecting **Tapes** on the **Home** tab. You must at least be connected to the DIVA Core database to access this view. The Tapes view provides flexible search criteria (*located at the top of the screen*) to execute DIVA Core database queries about the tapes managed by DIVA Core. You can execute search queries independently whether the Manager is running or not. However, the Control GUI must be connected to the DIVA Core database.

For Sony Optical Drives, you can view whether a media is Write-Once by clicking on the tape (*the Write-Once property is displayed in the Tape Properties window*). The Blu-ray discs are shown as tapes and viewable in the Tapes view panel of the Control GUI. The ***Write Once Media*** column displays this information as either **Y** or **N** indicating whether the tape is Write-Once. You can also filter the view so only Write-Once media is displayed.

Right-clicking on a specific tape in the Tapes view will produce a context menu with the additional options you can perform on the selected tape. The Tape Properties dialog is for informational purposes only. No data within the tape can be directly manipulated by an *operator* from this dialog box. Selecting the **Properties** menu item on the **Tapes** context menu displays the Tape Properties screen for the selected tape.

Tape Compression

Tape compression is supported at the tape group level.

When tape compression is enabled, any empty tape assigned to the group will have compression enabled, and instances written to the tape will be compressed. Tapes assigned to the group before compression was enabled remain uncompressed, and instances written to the uncompressed tape will be uncompressed.

To view all tapes with compression enabled, you must select the **Home**, **Tapes** icon in the Control GUI, and set the ***Compression*** filter to **Y**.

Tape Drive Encryption

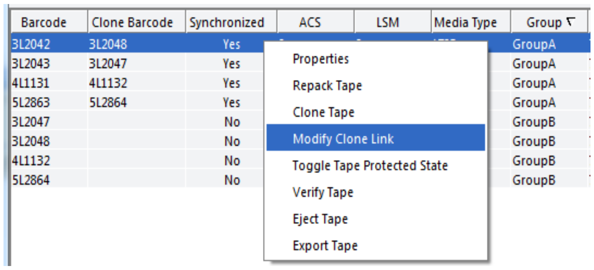
Tape drive encryption securely supports bulk tape migration between DIVA Core systems. Group level encryption is enabled, disabled, or updated in the *Groups* view of the **Sets, Groups & Media Mapping** tab in DIVA Command.

You can view the encryption status of the tape on the **Home**, **Tapes** screen in the Control GUI. The tape encryption status is also displayed on the *Tape Properties* screen when you double-click a tape in the **Tapes** view.

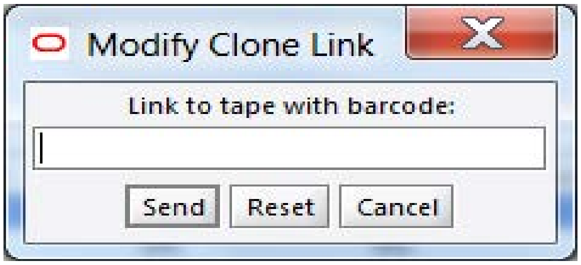
See the Installation and Configuration Guide for detailed configuration, and export and import information.

Modification of Clone Links

To modify a clone link (*set or delete the link*) between tapes, you must select the **Modify Clone Link** option from the **Tapes List** view in the Control GUI. Specify an empty barcode to delete the link, or a valid barcode to link the tape to another tape with the specified barcode.



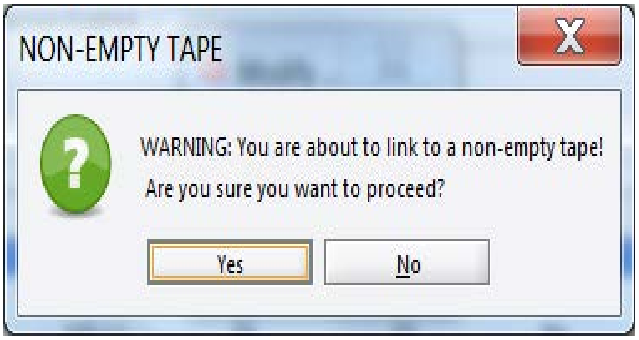
Barcode	Clone Barcode	Synchronized	ACS	LSM	Media Type	Group
3L2042	3L2048	Yes				GroupA
3L2043	3L2047	Yes				GroupA
4L1131	4L1132	Yes				GroupA
5L2863	5L2864	Yes				GroupA
3L2047		No				GroupB
3L2048		No				GroupB
4L1132		No				GroupB
5L2864		No				GroupB



Modify Clone Link

Link to tape with barcode:

Note: If a user attempts to link to a compatible non-empty tape, they will be prompted with a warning to verify that they really want to link to a non-empty tape as shown in the following figure. Two tapes are compatible if they have the same format, type, total size, and last written block.



NON-EMPTY TAPE

WARNING: You are about to link to a non-empty tape!

Are you sure you want to proceed?

On assignment of a clone link, the Clone Tape is set *Protected*. Removing the corresponding group link does not affect the set of already clone linked tapes.

Home Tab: Servers

The Control GUI Servers view provides information about the sources and destinations identified in the DIVA Core system. You view this display by clicking the **Source Destination Servers** button on the **Home** tab. This view displays the Source or Destination Server ***Name***, ***Product System***, ***Type***, ***Address***, and ***First Utilization Date***. Double-clicking one of the entries displays the Entry Details dialog box.

The information in the dialog box ***Core*** frame includes ***Name***, ***Type***, Network, ***Site ID***, and ***First Utilization Date*** fields.

The information in the dialog box ***Connection*** frame includes ***Address***, ***Root Path***, and ***Options*** fields.

The information in the dialog box ***Data Limits*** frame includes ***Max Throughput***, ***Max Accesses***, ***Max Read Accesses***, and ***Max Write Accesses*** fields.

The information in the dialog box is not editable and is only for informational purposes.

Action Tab

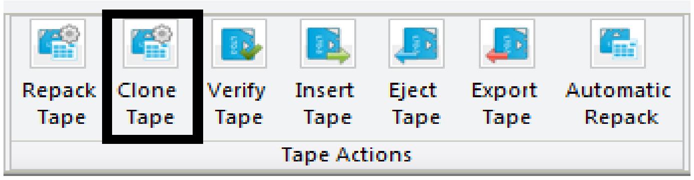
You perform most object-based operations (***Archive, Restore, Copy, Delete****, and so on*) from the **Action** tab on the ribbon bar.

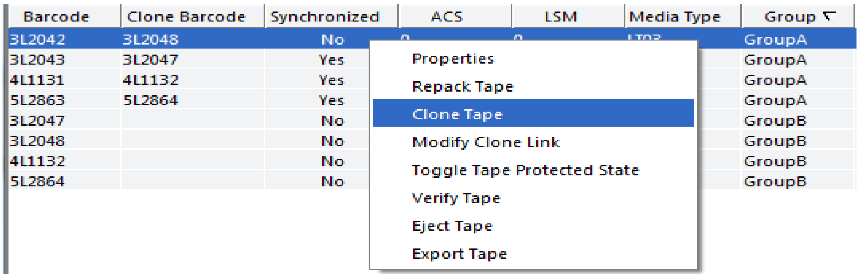
When restoring the same file to the same destination twice in parallel, the behavior on Windows and Linux is different. On Windows, the first restore (*they cannot arrive exactly at the same time*) will lock the file so that the second one will terminate. On Linux, there is no such lock at the file system level. Both restores are executed at the same time, and both will write to the same file. The content of the resultant file is not predictable.

On the right side of the **Action** tab is the **Tape Actions** button. You click the **Tape Actions** button to perform various tape operations. Most tape operations are self-explanatory. However the **Automatic Repack** operation requires some description and is covered in the following section.

Manual Tape Cloning

Users either select **Clone Tape** from the Control GUI's **Tape Actions** bar, or right-click a Source Tape from the **Tapes** view the list of actions displayed to clone a tape.

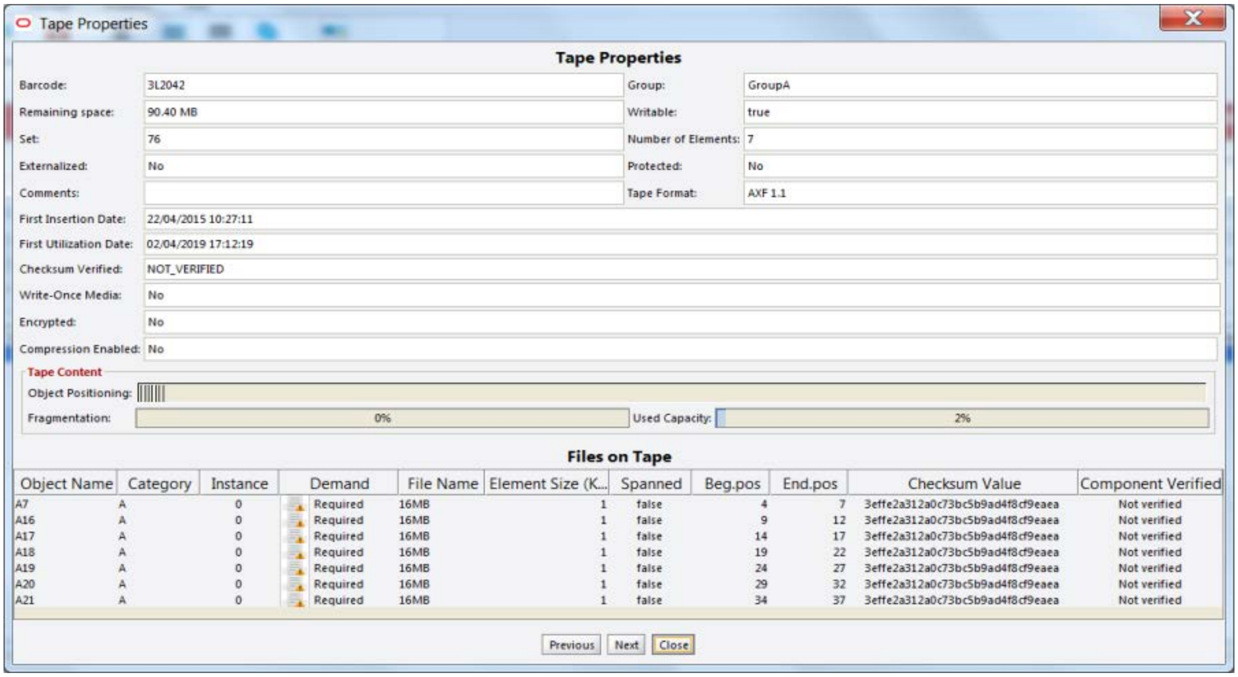




Barcode	Clone Barcode	Synchronized	ACS	LSM	Media Type	Group
3L2042	3L2048	No	0	0	LTO3	GroupA
3L2043	3L2047	Yes				GroupA
4L1131	4L1132	Yes				GroupA
5L2863	5L2864	Yes				GroupA
3L2047		No				GroupB
3L2048		No				GroupB
4L1132		No				GroupB
5L2864		No				GroupB

After one or more archives to a group containing a Clone Group, users can manually clone the tape containing the newly archived objects. For example, upon archiving of 6 objects to a group called *GroupA* configured with a Clone Group named *GroupB*, a user can clone the tape containing the newly archived objects using one of the aforementioned methods.

Users can view the objects on the Source Tape and Clone Tapes before actually cloning the tape as shown in the following figures. The first figure shows the objects on the Source Tape and the second figure shows the objects on the Clone Tape.



Tape Properties

Barcode:	3L2042	Group:	GroupA
Remaining space:	90.40 MB	Writable:	true
Set:	76	Number of Elements:	7
Externalized:	No	Protected:	No
Comments:		Tape Format:	AXF 1.1
First Insertion Date:	22/04/2015 10:27:11		
First Utilization Date:	02/04/2019 17:12:19		
Checksum Verified:	NOT_VERIFIED		
Write-Once Media:	No		
Encrypted:	No		
Compression Enabled:	No		

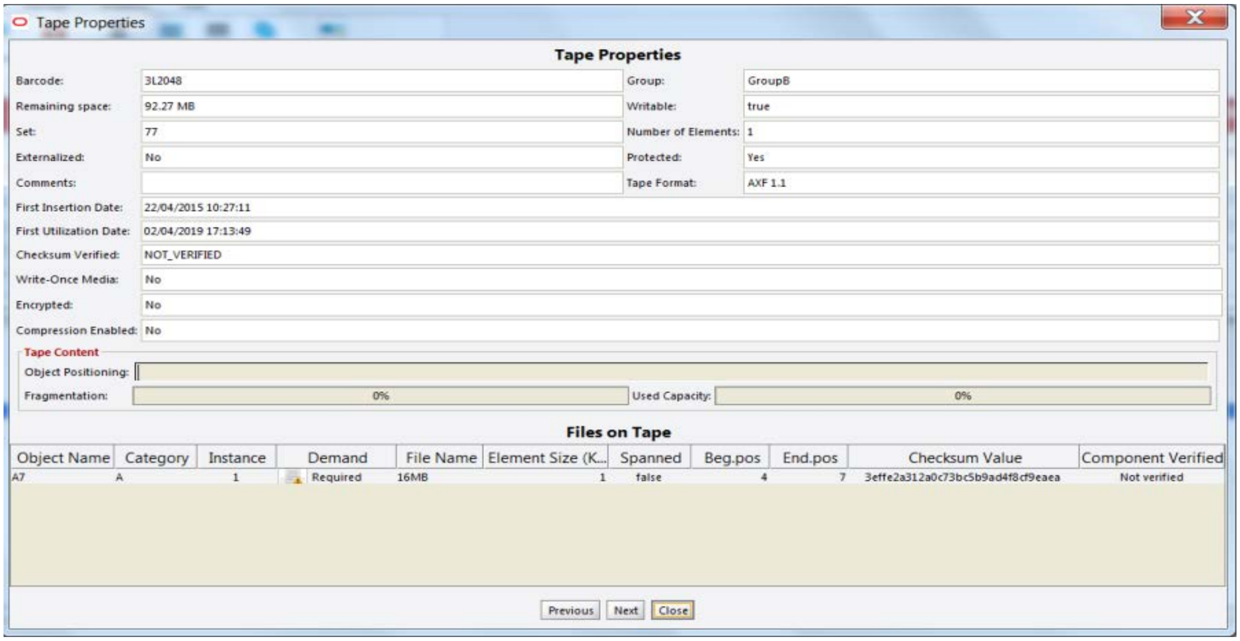
Tape Content

Object Positioning:

Fragmentation: 0% Used Capacity: 2%

Files on Tape

Object Name	Category	Instance	Demand	File Name	Element Size (K...	Spanned	Beg.pos	End.pos	Checksum Value	Component Verified
A7	A	0	Required	16MB	1	false	4	7	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A16	A	0	Required	16MB	1	false	9	12	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A17	A	0	Required	16MB	1	false	14	17	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A18	A	0	Required	16MB	1	false	19	22	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A19	A	0	Required	16MB	1	false	24	27	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A20	A	0	Required	16MB	1	false	29	32	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A21	A	0	Required	16MB	1	false	34	37	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified



Tape Properties

Barcode:	3L2048	Group:	GroupB
Remaining space:	92.27 MB	Writable:	true
Set:	77	Number of Elements:	1
Externalized:	No	Protected:	Yes
Comments:		Tape Format:	AXF 1.1
First Insertion Date:	22/04/2015 10:27:11		
First Utilization Date:	02/04/2019 17:13:49		
Checksum Verified:	NOT_VERIFIED		
Write-Once Media:	No		
Encrypted:	No		
Compression Enabled:	No		

Tape Content

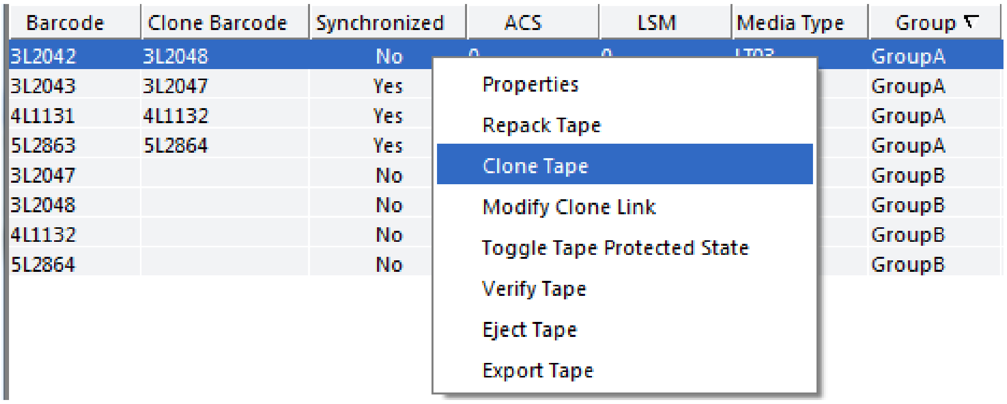
Object Positioning:

Fragmentation: 0% Used Capacity: 0%

Files on Tape

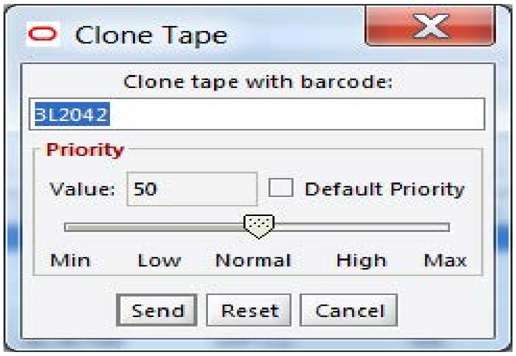
Object Name	Category	Instance	Demand	File Name	Element Size (K...	Spanned	Beg.pos	End.pos	Checksum Value	Component Verified
A7	A	1	Required	16MB	1	false	4	7	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified

Comparing the objects on each tape in the previous figures, the Clone Tape (*the second figure*) is missing 6 objects (*A16 through A21*). The Source Tape is also not synchronized with its clone. You can see this by viewing the new *Synchronized* flag of the Source Tape in the **Tape List View**. To exclusively copy the missing content from Source Tape 3L2042 to Destination Tape 3L2048, right-click on the Source Tape containing the newly archived objects, then select **Clone Tape**, and submit the request.



Barcode	Clone Barcode	Synchronized	ACS	LSM	Media Type	Group
3L2042	3L2048	No	0	0	LT03	GroupA
3L2043	3L2047	Yes				GroupA
4L1131	4L1132	Yes				GroupA
5L2863	5L2864	Yes				GroupA
3L2047		No				GroupB
3L2048		No				GroupB
4L1132		No				GroupB
5L2864		No				GroupB

After selecting the Clone Tape menu option the following dialog box is displayed. Click Send to clone the tape.



Clone Tape

Clone tape with barcode: 3L2042

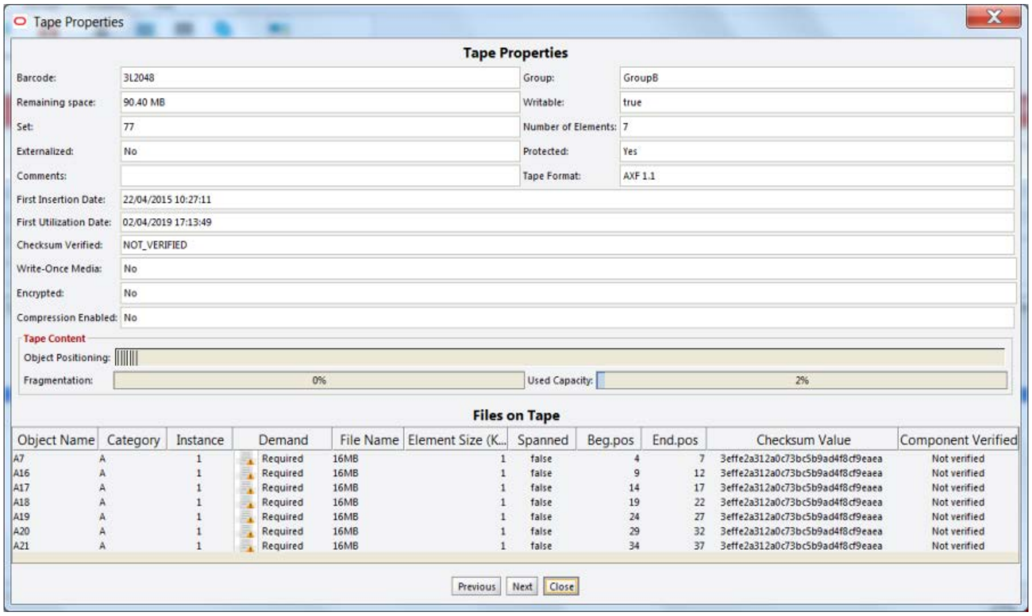
Priority

Value: 50 □ Default Priority

During the first clone of a Source Tape, an empty tape contained in the set associated with the Clone Group is selected. The selected tape will be of the same type as the Source Tape, and equal to or greater in size. The newly selected Clone Tape is marked as *protected*.

Only clone requests can write to a Clone Tape, and any attempts to write to the Clone Tape using any other request will terminate. Every subsequent clone of the same Source Tape will write to the same Clone Tape. The format of the Clone Tape will be the same as that of the Source Tape.

You can verify that the Clone Tape contains the same content as the Source Tape by viewing the Clone Tape after the clone request completes. The remaining space, number of elements, objects and positions all match the Source Tape.



Tape Properties

Tape Properties

Barcode:	3L2048	Group:	GroupB
Remaining space:	90.40 MB	Writable:	true
Set:	77	Number of Elements:	7
Externalized:	No	Protected:	Yes
Comments:		Tape Format:	AXF 1.1
First Insertion Date:	22/04/2015 10:27:11		
First Utilization Date:	02/04/2019 17:13:49		
Checksum Verified:	NOT_VERIFIED		
Write-Once Media:	No		
Encrypted:	No		
Compression Enabled:	No		

Tape Content

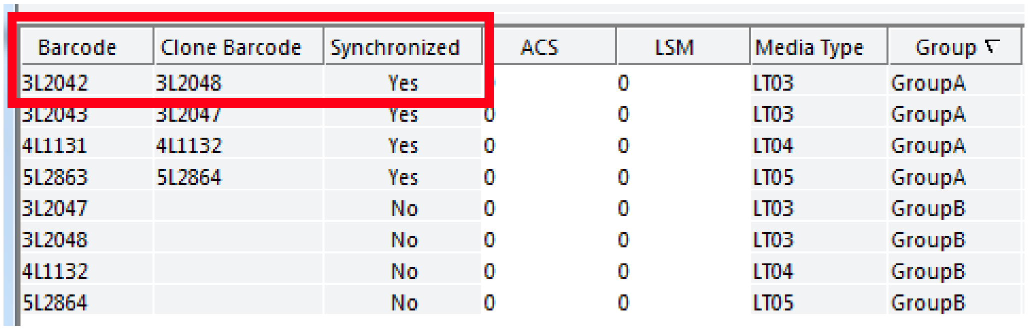
Object Positioning:

Fragmentation: 0% Used Capacity: 2%

Files on Tape

Object Name	Category	Instance	Demand	File Name	Element Size (K...	Spanned	Beg.pos	End.pos	Checksum Value	Component Verified
A7	A	1	Required	16MB	1	false	4	7	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A16	A	1	Required	16MB	1	false	9	12	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A17	A	1	Required	16MB	1	false	14	17	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A18	A	1	Required	16MB	1	false	19	22	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A19	A	1	Required	16MB	1	false	24	27	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A20	A	1	Required	16MB	1	false	29	32	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified
A21	A	1	Required	16MB	1	false	34	37	3effe2a312a0c73bc5b9ad4f8cf9eaea	Not verified

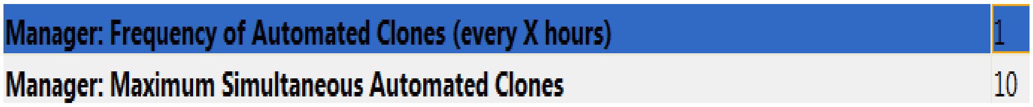
Also, the *Synchronized* state of the Source Tape is updated to reflect that the source tape is now synchronized with its clone.



Barcode	Clone Barcode	Synchronized	ACS	LSM	Media Type	Group
3L2042	3L2048	Yes		0	LT03	GroupA
3L2043	3L2047	Yes	0	0	LT03	GroupA
4L1131	4L1132	Yes	0	0	LT04	GroupA
5L2863	5L2864	Yes	0	0	LT05	GroupA
3L2047		No	0	0	LT03	GroupB
3L2048		No	0	0	LT03	GroupB
4L1132		No	0	0	LT04	GroupB
5L2864		No	0	0	LT05	GroupB

Automated Cloning

In addition to manually cloning a tape, users can set up the periodic cloning of all tape groups configured for cloning by configuring two settings in the **Settings** screen of DIVA Command. The first setting regulates the frequency of the automated clones; the value identifies the number of hours between the submission of clone requests. The second parameter determines the maximum number of simultaneous clones. In the following figure the user configured the periodic cloning of a maximum of 10 tapes every hour. Tapes are cloned in order of least recently cloned. A clone request is only attempted if objects were written to the tape after the last clone.



Manager: Frequency of Automated Clones (every X hours) 1

Manager: Maximum Simultaneous Automated Clones 10

Automatic Repack

When DIVA Core writes an object to a tape, the object can only be appended to where the last object was written on that tape. When any object is subsequently deleted from a tape, the space from that object cannot be reused. Eventually, as more and more objects are deleted, tape fragmentation occurs and potentially develops a considerable amount of unusable space in the tape library.

This unusable space can be reclaimed by repacking the tape. The repack process reads all material from the tape being repacked to a temporary cache and then writes it back to a new tape in the same group as the original (*sequentially*).

You can perform a manual tape repack by clicking the **Repack Tape** button, or triggered automatically when tape fragmentation and used capacity thresholds are exceeded.

You enable or disable **Automatic Repack** by clicking the **Automatic Repack** button on the **Action** tab.

Note: **Automatic Repack** ignores WORM Media. If WORM Media is repacked manually, the space is not recoverable.

Clicking the **Automatic Repack** button displays a dialog box. To enable **Automatic Repack** select the check box, and then click the **Send** button. To disable it, deselect the check box, and then click the **Send** button.

After enabling **Automatic Repack**, a second dialog box is displayed for you to configure the repack parameters including ***Tape Filling Threshold (%)***, ***Tape fragmentation threshold (%)***, ***Start Time***, ***Duration***, and ***Repack tapes from selected group***.

OTU (*Object Transfer Utility*) for Cloud Source and Destination Servers

The first icon on the **Actions** tab is for **OTU** (*Object Transfer Utility*). After an OPC cloud destination is defined, you can use OTU to browse through the container directory tree for objects in an Oracle Object Storage Account. OTU loads the immediate children of the container folder, and additional files and folders are added to the tree dynamically.

You can also use OTU to restore content to a container and folder of an object in an Oracle Object Storage Account destination, or archive an object from an Oracle Object Storage Account source to DIVA Core.

The Cloud OTU can also identify a manifest file and remove all file fragments included in the manifest file so that a single file is displayed.

See the OTU (Object Transfer Utility) User's Guide for configuration and operational details.

Manage Tab: Objects

You display this view by clicking **Objects** on the **Manage** tab. You must at least be connected to the DIVA Core Database to access this view. You can search objects

You select an object and right-click it to display the objects context menu. These commands are a subset of those from the **Action** tab menu and perform the same functions. At this context level, the request window (*by default*) does not automatically specify an instance of the object in the ***Instance*** field of the request. You can manually enter a specific instance in this field before the command is sent. If the ***Instance*** field is left blank, the command will deal with all instances of the object. For example, if you select **Delete** from this menu, and no instance number is specified in the request, DIVA Core will delete *all* instances of the object.

The Object Properties dialog box displays an object's properties, instances, and components. A valid instance number must be specified in any command issued from this view or you will receive an error message when submitting the command.

This includes removing the instance number entirely from the request. For example, you cannot delete all instances of an object from this menu by leaving the ***Instance*** field empty.

This view will not permit the last instance of the object to be deleted and DIVA Core will automatically terminate this request.

If a *file* (*or part thereof*) of an object instance is spanned across two or more tapes, and only one tape of the set is externalized, the instance is still considered externalized. However, an *object* (*that is, all instances*) is only considered externalized if all instances of that object are externalized.

Clicking on the **Instances** tab in the bottom section of the Object Properties screen displays the object's Instances screen. This view displays the instances of the object and the elements of each instance.

Clicking on the **Components** tab in the bottom section Object Properties screen produces the Object Component screen. Clicking on a displayed component will display the component's information at the bottom of the window.

Manage Tab: Requests

You display this view by clicking on **Requests** on the **Manage** tab. This view is limited to 300 lines by default. Completed, canceled, and aborted requests are cleared if the Control GUI is disconnected or relaunched. The **Requests** view is provided for retrieval of previously completed requests from the DIVA Core database. It is commonly used by Telestream Support to troubleshoot a previously reported issue.

Errors can be identified quickly by clicking the ***Errors*** column. Once a particular request's ***Request ID*** is determined, it can be entered in the **Events** view to export the request's event log from the Control GUI to a text file. You can search the displayed requests using the search area at the top of the screen.

Double-clicking on a request displays the Request Properties screen for viewing information (*including Request Properties, Object Properties, Archive Properties and Events List*) about the selected request. Up to fifty thousand logged requests are stored in the DIVA Core database. Once this number of requests is reached, the oldest requests are overwritten. For most facilities, this provides at least approximately six to twelve months of logged requests.

Manage Tab: Media

You display this view by clicking on **Media** on the **Manage** tab. This view displays information for each of the tape groups and disk arrays identified in the DIVA Core system. You filter the search using the lists at the top of the screen.

The ***Name*** list can be a full or partial media name including wildcards. An asterisk will display all media names.

You use the Type list to select viewing **All** media types, only **Groups**, or only **Arrays**.

You click **Refresh** after making your filtering selections to update the display.

Double-clicking on the tape group or disk array displays a dialog box with details about that group or array. The screen is for informational purposes only and is not editable.

Source Media Priority

The ***Source Media Priority*** determines which source instance is preferred (*according to the media where the instance resides*) during the instance selection process of a Restore, Partial File Restore, and Copy To Group request. Instances on media with a higher priority are preferred. Cloud instances are only copied or restored if all local instances are offline, or no local instances exist. This is an absolute condition independent of the ***Source Media Priority***.

Manage Tab: Require/Release

You display this view by clicking on **Require/Release** on the **Manage** tab. You must at least be connected to the DIVA Core database to access this view. This view allows an *operator* to query the DIVA Core database for the following:

- Released instances that can be externalized from any libraries managed by DIVA Core.
- Externalized instances that are required to be inserted into a library.
- A list of tapes that must be inserted to fulfill Restore requests (*required tapes*).
- A list of tapes you can externalize from a library (*releasable tapes*).

Note: A releasable tape only contains released instances.

You use the filters at the top of the screen to limit the type and number of results returned from your query. You run your query by clicking **Refresh** on the top right of the screen. There are only two filters available as follows:

Dates

Select the ***Begin*** and ***End*** dates and times to search. Select the ***Enable*** check box to enable this option for the query. Deselect the check box to not include this option in the search.

Demand

You use this list to select viewing either **Required & Externalized** tapes (*only required tapes that are externalized are displayed*), or **Released & Inserted** tapes (*only released tapes that are internalized and inserted are displayed*).

At the bottom of the screen are the **Required Tapes** button and the **Releasable Tapes** button. The **Required Tapes** button generates a list of tapes required to be inserted into the library. The **Releasable Tapes** button generates a list of tapes that have their instances released, and can be externalized from the library.

Clicking either button displays the associated dialog box. The Required Tapes screen enables you to view, print, or save the list to a text file. The Releasable Tapes screen enables you to view, print, save the list to a text file, or eject the tape by clicking the **Eject Tape** button.

Manage Tab: SPM Actions

You display this view by clicking on **SPM Actions** on the **Manage** tab. You must at least be connected to the DIVA Core database to access this view. This view is only applicable to installations having SPM (*Storage Plan Manager*) installed. It enables more detailed information to be extracted from the DIVA Core database related to the actions that have been initiated to DIVA Core from the SPM module.

You use the filters at the top of the screen to limit the type and number of results returned from your query. You run your query by clicking **Refresh** on the top right of the screen.

Right-clicking on a result returned by the SPM Actions query displays the SPM Actions Context menu. The menu has only two options as follows:

Request Properties

This displays the request's events list associated with the SPM Action. This is only applicable to SPM Actions where the ***Request ID*** is nonzero.

Reschedule Action

If the SPM-initiated request failed (*for example, the medium or associated Datahubs for the slot were unavailable*), this enables you to retry that SPM Action.

Analytics Tab: Metrics

You display this screen by clicking on **Metrics** on the **Analytics** tab. The Metrics view provides a set of filters to narrow your searches. Information can be filtered by ***Metric Definition***, ***Collection Interval***, ***Aggregation Item***, ***Resource Name***, ***Value***, ***Count***, ***Start Date***, and ***Last Update Date***. The ***Metric Definition*** list contains the metrics defined in DIVA Command plus the built-in ones (*DIVAPROTECT**).

See the *DIVAprotect User's Guide* for more information.

Analytics Tab: Events

You display this screen by clicking **Events** in the **Analytics** tab. You must at least be connected to the DIVA Core database to access this view. The Events view is typically used with the Requests view for troubleshooting purposes. You can filter the displayed results using the filters at the top of the screen including ***Dates*** (***Start*** *and* ***End*** *dates and times*), ***Severity*** (***Information****,* ***Warnings****,* ***Errors****, and* ***Critical***), ***Request ID***, and ***Description***.

When a particular request fails, you can export the log of that request to a text file and send it to Telestream Support (*when requested*). This information can also be collected directly by the Telestream Support Engineer using the *Customer Information Collection Tool*.

When the query is run for the failed request's ***Request ID*** (*usually retrieved from the Requests view*), it shows the same events of that request's event log. You can save this file as a text file by selecting **Export**.

DIVA Core stores a maximum of one million events in its database. When the number of logged events exceeds this value, DIVA Core will begin overwriting the existing events beginning with the oldest entry.

Analytics Tab: Drive Alert Logs

You display this screen by clicking **Drive Alert Logs** in the **Analytics** tab. This view lists errors reported by tape drives. This information is vendor-specific and may vary depending on the make and model. You can filter your search using the filters at the top of the screen. For example, you can look for errors related to a particular tape.

Analytics Tab: Library Alert Logs

You display this screen by clicking **Library Alert Logs** in the **Analytics** tab. This view lists errors reported by direct-attached SCSI protocol libraries. This information is vendor-specific and may vary depending on the library make and model. You can filter your search using the filters at the top of the screen.

Analytics Tab: DIVA Core Information

You display this screen by clicking **DIVA Core Information** in the **Analytics** tab. The amount of storage that DIVA Core will manage is dictated by the DIVA Core configuration. When the total managed capacity reaches ninety percent of the capacity, DIVA Core will begin issuing periodic warning messages in the **Manager Requests** view of the Control GUI.

The warning messages become more frequent when the managed capacity meets the configured capacity. When this limit is reached, no further Archive requests are accepted by DIVA Core and they will be automatically terminated. However, Restore requests will continue to be accepted.

Analytics Tab: Database Logs

You display this screen by clicking **DIVA Core Information** in the **Analytics** tab. This view lists errors reported to the Oracle database. A set of filters is available to narrow down searches. For example, you can look for errors related to a particular tape.

View Tab: Properties, Clear, Clear All

You use the **View** tab to view the properties of selected items in other Control GUI screens. You highlight the items (*object, system component, and so on*) in the original screen, click the **View** tab, and then click the **Properties** button.

The **Clear** and **Clear All** buttons on the **View** tab clear one or more requests, errors, warnings, and so on from the appropriate screens.

Exporting the Current View

To export the information currently displayed in almost any Control GUI view, click the **Start** orb, and then click **Export Current View** from the **Tools** menu.

Exporting is available for the Home Tab: Tapes view and all other views in the Control GUI *except* the following tables:

- Home Tab: Manager (Current Requests View)
- Home Tab: Datahubs
- Home Tab: Robot Managers
- Home Tab: Libraries
- Home Tab: Drives
- Home Tab: Disks
- Home Tab: Servers
- Action Tab
- Manage Tab: Requests

In the file save dialog box that appears, select an existing file or specify the name of a new file that will receive the information in the currently displayed table. Use the following procedure to save, and then view, the events in a spreadsheet such as Excel:

1. Enter the .csv extension after your file name in the ***File Name*** field.
2. Click **Export** to complete the export of the view.

The content of the exported view will depend upon the current filter selection. If the filters have been modified since the last query, and you have not clicked **Refresh**, the exported log will not represent the current filter selections. Also, if the table in the Detailed View is empty when the view is exported, the destination file will contain nothing except two lines indicating the current filter selections.

7

Removable Media Operations

This chapter describes the tape exporting and importing procedures, and includes the following information:

Export/Import Overview

The *Export* function (*on the first DIVA Core site*) generates metadata files that describe each tape selected for export, and then ejects the selected tapes from their current tape library.

You use the *Import* function to import the metadata, and then insert the ejected tapes into the second system. The archived objects on the exported tapes are then transferred to the second DIVA Core system.

All export functions and the Insert Tape command are executed from the Control GUI. The Import Tape function uses the command-line interface. DIVA Core enables more than one set of tapes (*whether spanned or not*) to be exported to and imported from a single file.

Newly imported objects will have only one instance - the instance residing on the tape(*s*) that was imported. You also have the option to import an object as an instance of another object already existing in the DIVA Core database. The Import Utility requires your specification of a target tape group for newly imported tape objects. The new objects will belong to the identified tape group and not the tape group of the DIVA Core system from which it was exported.

The Export/Import functionality is compatible with complex objects and has additional fields for the advanced formatting and functionality available in DIVA Core release 8.1.

Note: The exported metadata from a DIVA Core 8.1 export cannot be imported into DIVA Core releases before 7.0. However, exported metadata created from releases of DIVA Core before 8.1 can be imported into the DIVA Core 8.1 system.

Tape Drive Encryption

Starting with the DIVA Core 8.0 release, tape drive encryption supports secure bulk tape migration between DIVA Core systems.

After enabling encryption on a tape group, all additional tapes added to the group will also be encrypted. However, any existing tapes in the group remain unencrypted if encryption was previously disabled.

Enabling encryption on a tape group generates an encryption key, which is also encrypted. You can change the encryption key at any time from DIVA Command. Updating the encryption generates a new key.

New tapes added to the group after the change will use the new encryption key. The existing tapes that were already encrypted will continue to use the original key. Therefore, tapes in the same tape group can have different encryption keys. You must notify the Manager of the change when updating the encryption key.

Disabling encryption (*after it is already enabled*) only affects additional tapes added to the group, and the existing tapes remain encrypted.

You can view the encryption status of the tape on the **Home**, **Tapes** screen in the Control GUI.

See the Installation and Configuration Guide for detailed configuration information.

Tape Compression

Tape compression is supported at the tape group level, and configured in DIVA Command.

When tape compression is enabled, any empty tape assigned to the group will have compression enabled, and instances written to the tape will be compressed. Tapes assigned to the group before compression was enabled remain uncompressed, and instances written to the uncompressed tape will be uncompressed.

When exporting a tape, compression is tracked using the new `isCompressionEnabled` attribute. This attribute value can be either `true` or `false`.

To view all tapes with compression enabled, you must select the **Home**, **Tapes** icon in the Control GUI, and set the ***Compression*** filter to **Y**.

Exporting and Importing through the Java API

You can now export and import tapes through the Java API, and also in the Java Initiator. The following is sample output from the export and import of a single encrypted tape using the Java Initiator.

```
DIVA Core JInitiator - Using JavaAPI Version : 8.1 SNAPSHOT

0 = Exit
```

```
1 = Connect

2 = Archive object
3 = Copy to New object
4 = Copy to Group (new instance)
5 = Associative Copy

6 = Delete Object
7 = Delete Instance
8 = Delete File
9 = Delete File For Category Mask

10 = Restore Object
11 = Restore Instance
12 = N - Restore
13 = Partial Restore

14 = TranscodeArchive
15 = Transfer

16 = Insert tape
17 = Eject tape

18 = Cancel Request
19 = Change Priority
20 = Get Request Information
21 = Get Partial Restore Request Information
22 = Get Finished Request List

23 = Get Object Info
24 = Get Tape Info
25 = Get Object Details List
26 = Get Files and Folder Names for Object

27 = Get Array List
28 = Get Server List
29 = Get Tape Group List

30 = Add Tape Group
31 = Delete Tape Group
32 = Require Instance
33 = Release Instance

34 = Get Storage Plan Names List
35 = Get Object List By File Name
36 = Get Archive System Information

37 = Lock Object
38 = Unlock Object
39 = Link Objects
40 = Enable Automatic Repack
41 = Export Tapes
42 = Import Tapes

100= close connection

Enter a command number : 41
*** exportTapes ***
Specify the Tape barcodes.
```

```
Barcode 1 <1S0009>: 3L2247

Add another? <N>:

Comment <>:

Set delete from database (Y/N) <Y>:

Priority <-1>:


--- Export Tapes ---
Tapes
3L2247
Comment:
Delete from database: true
Priority: -1


Submit[S] or Submit and Wait[W] <S>:

Status: Running
Request ID: 20272

Success

Press Enter to continue.

Request 20272 has completed successfully


DIVA Core JInitiator - Using JavaAPI Version : 8.1 SNAPSHOT

0 = Exit
1 = Connect

2 = Archive object
3 = Copy to New object
4 = Copy to Group (new instance)
5 = Associative Copy

6 = Delete Object
7 = Delete Instance
8 = Delete File
9 = Delete File For Category Mask

10 = Restore Object
11 = Restore Instance
12 = N - Restore
13 = Partial Restore

14 = TranscodeArchive
15 = Transfer

16 = Insert tape
17 = Eject tape

18 = Cancel Request
19 = Change Priority
20 = Get Request Information
```

```
21 = Get Partial Restore Request Information
22 = Get Finished Request List

23 = Get Object Info
24 = Get Tape Info
25 = Get Object Details List
26 = Get Files and Folder Names for Object

27 = Get Array List
28 = Get Server List
29 = Get Tape Group List

30 = Add Tape Group
31 = Delete Tape Group
32 = Require Instance
33 = Release Instance

34 = Get Storage Plan Names List
35 = Get Object List By File Name
36 = Get Archive System Information

37 = Lock Object
38 = Unlock Object
39 = Link Objects
40 = Enable Automatic Repack
41 = Export Tapes
42 = Import Tapes

100= close connection

Enter a command number : 42
*** importTapes ***
Specify the file or directories to import

File 1 <>: D:\workspace\Manager\bin\exported\2017-04-04--17.28.57

Add another file? <N>:

Group name <>: default

Skip Object import if already exists in database (Y/N) <N>:

Add as instance if the object already exists in database (Y/N) <N>:

--- Import Tapes ---
Files and Directories
D:\workspace\Manager\bin\exported\2017-04-04--17.28.57
Group name: default


Continue? <Y>:


---------- Response ----------
Success

Press Enter to continue.
```

Exporting Tapes

The *Export Tapes* function enables one or more tapes containing objects to be exported for use in another independent DIVA Core system (*for example at a remote disaster recovery or partner site*).

The metadata of each tape for non-complex objects are maintained in the DIVA Core database. The metadata of each tape is saved to an XML file when the tape(*s*) are exported and used to transfer the metadata to the other DIVA Core system's database during the import operation.

The metadata for complex objects is maintained in both the DIVA Core database and the metadata database. When an export request is initiated, the Export Utility creates an additional plain text file and assigns a .ffm extension to the file.

The export feature checks to see if any of the selected tapes contain objects that span onto other tapes. If so, these tapes are included in a menu so that they can also be exported. These spanned tapes must be selected to export the original list of tapes.

The Export Tapes command is not used for transferring tapes between two or more libraries controlled by the same Manager. To transfer tapes between libraries under the same Manager's control, you use the Eject command, move the tape to the desired library, and then execute an Insert Tape command.

The default action in the export feature removes the tape metadata from the DIVA Core database after the export. In this case, if an object being exported is the last (*or only*) instance of the object, it will be removed entirely from the database. However, the object metadata can be left in the original DIVA Core database if desired.

Ejected tapes can also be exported. Ejecting tapes before exporting them is the recommended method when the number of tapes to be exported exceeds the robotic tape library selected CAP (Cartridge Access Port) size.

The media type (*Write-Once or not*) and whether the media is a cartridge or not is identified in the exported XML file and also imported during an Export/Import operation. The new attributes of the tape element are isWriteOnce and isCatridge each with a value of either true or false.

Export Limitations

Tape export limits are configured in the manager.conf configuration file. There are several configurable parameters as described in the following table.

Table 7–1 Tape Export Limitation Parameters

Parameter	Definition	Limits
DIVAMANAGER_MAX_EXPORT_TAPES	The maximum number of tapes allowed in an export request. Reloadable in *SERVICE* mode.	The range of possible values is 1 to 1000. Example: DIVAMANAGER_MAX_EXPORT_TAPES=10
DIVAMANAGER_MAX_EXPORT_ELEMENTS	The maximum number of elements allowed in an export request. Reloadable in *SERVICE* mode.	The range of possible values is 1 to 10000000. Example: DIVAMANAGER_MAX_EXPORT_ELEMENTS=1000000

Telestream *highly* recommends:

- Only performing one export operation at a time. You risk data loss if more than one export operation is running simultaneously.

- Not performing large exports during peak periods. System performance will be decreased during large exports.
- Delete and repack actions do not clear WORM drives as these are Write-Once Media. The instances are deleted but the space is not recoverable.

Exporting Encrypted Tapes

An encryption key hash and salt are generated during the export of encrypted tapes. This key hash and key salt are stored in the metadata file. The export process optionally generates a new password protected Keystore file in the same folder as the metadata file. You configure this setting on the **Manager Settings** tab in DIVA Command. You must be in Engineering mode to access the setting.

The Keystore file contains information used to import encrypted tapes.

Note: A valid Keystore password is required to export encrypted tapes. See the following section for information.

Export Keystore

You can verify the integrity of the Keystore file using the Java keytool. See https://docs.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html for details on Java's keytool.

The Keystore password is set in DIVA Command in the *Manager Configuration* view. You enter the Keystore password in the ***Export: Tape Encryption Keystore Password*** field. The password must be at least eight characters and contain at least one digit, at least one lowercase alphabetic character, at least one upper case alphabetic character, and at least one special character within a set of special chars (*!@#%$^*).

You enable exporting encryption keys by selecting the ***Export: Enable Export of Encryption Keys*** check box. Exporting encryption keys is disabled by default. You must be in Engineering mode to view or edit both settings.

Export Metadata Parameters

The following table describes the export metadata parameters.

Table 7–2 Export Metadata Parameters

Parameter	XML Element and Attribute	Notes
objectId	Attribute of the object element	Not imported - A new Object ID is generated during import.
uuid	Attribute of the object element	Imported if present, otherwise a new UUID will be generated.
format	Attribute of the object element and attribute of the tape element	0 = Legacy 1 = AXF 0.9 2 = AXF 1.0 -1 = Unknown
numFolders	Attribute of the object element	
isHeaderValid	Attribute of the object element	
isComplex	Attribute of the object element	

Table 7–2 (Cont.) Export Metadata Parameters

Parameter	XML Element and Attribute	Notes
footerBeginPos	Attribute of the element's element	If exists in the database
footerEndPos	Attribute of the element's element	If exists in the database
compOrderNumBegin	Attribute of the element's element	If exists in the database
compOrderNumEnd	Attribute of the element's element	If exists in the database
fileFolderMetadataInfo	Element	Valid for complex objects
fileFolderMetadataInfo-elem	Element	Valid for complex objects
checksums and checksum	Element	Not valid for complex objects
elementIds	Attribute of the component element	The fully qualified path of Element ID values for a file or an empty folder's fully qualified path.

type	Attribute of the component element	Represents the type of object component:
		D = Directory
		F = File
		S = Symbolic Link in Linux
		Components of non-complex objects created before the 7.4 release default to F because only files were stored in non-complex objects before release 7.4.

Exported Tape Metadata Files

When tapes are exported from the DIVA Core system, DIVA Core writes each tape's metadata to a .xml file. DIVA Core generates an additional .ffm file for each exported complex object. If an object is spanned across two (*or more*) tapes, the XML file will encompass every tape included in the spanned set. The naming format of each tape metadata XML file is Tapeset-<Barcode>.xml (*for example Tapeset-000131.xml*).

The *Root Path* where the XML files are saved is defined by the DIVAMANAGER_EXPORT_ROOT_DIR parameter in the Manager configuration file. By default the export absolute folder *Root Path* is DIVA_HOME\Program\Manager\bin\exported\.

From this *Root Path* the .xml and .ffm files (*if complex objects exist*) from each Export Tapes command are saved in sub-directories based on the date and time the command was run.

The .ffm file contains file and folder information for complex objects. The .ffm files are referenced from within the specified .xml file and are named using the *Object Name* and *Object Category* of the exported object. This file must exist in the same directory as the .xml file when importing. The Import Utility will look for them both in the same location. If the file is missing, the import process will terminate and an error message will be written to the log file.

Export Tapes Procedure

Caution: When using complex objects, the FFM files must be in the same folder as the XML files for importing. If the FFM files are not found the import process will terminate and an error will be written to the log file.

The Export Tape request is initiated using the **Export Tape** button on the GUI ribbon bar, or the **Tapes** view in the **Home** tab by right-clicking the tape to export and selecting **Export Tape** from the resulting menu. When selecting the tapes for export, it is possible to see more tapes available in the tape window than initially selected. If a tape has objects that are spanned onto another tape, these tapes are also included. In this case, select all of the spanned tapes from this list for the export to succeed.

Use the following procedure to export tapes:

1. Highlight and then right-click the tapes desired for export.
2. Select **Export Tape** from the context menu to begin the export process.

   The Export Tape dialog box will appear showing information about the selected tapes and options for the export process. The available options include:

   Comments
   Enter any comments desired in the text box. They will be stored in the request's properties.

   Delete From DB
   If checked, the barcodes, tapes, and object instances stored on those tapes will be deleted from the DIVA Core database upon completion of the export. This parameter is set to true by default.

   If tapes or object instances are needed in the system again after they have been exported, you must import them because this option removes them from the system's database.

   Exported Tapes
   This area identifies which tapes were selected from the Control GUI for export, if the tape has the original barcode, and if it can be removed from the export operation. For example, if a tape is part of a tape set (*rather than a single tape*), the ***Can Be Removed*** column would indicate No for that tape because it is required to complete the export successfully.

   Remove Selected
   Removes the highlighted tapes in the **Exported Tapes** area from the export process.
3. After all options have been set and verified, click **OK** to begin the tape export.

   This is a multi-step process. If a set of tapes was selected that includes another spanned tape, the GUI will display re-selection dialogs enabling selection of additional tapes in the set.

   When the **OK** button is clicked, the export process begins. This results in a .xml (*and possibly .ffm files*) being created in the export folder. The XML and FFM files contain all of the information concerning the objects on the tape(*s*) being exported.

When the export is complete, a good practice is to compress all of the resulting files into a .zip file. You must include all of the files because they are required for the import process to complete successfully.

Bulk Tape Export

DIVA Core internally splits an export request into smaller exports consisting of no more than 100 tapes and 1 million tape elements. This process enables DIVA Core to accept a larger number of tapes and tape elements than can be handled by earlier DIVA Core releases.

The number of tapes an export request is split into will always be 100, and the number of tape elements will always be 1 million. These values are adjustable.

For example, if you set the maximum number of tapes value to 3, the resulting export is split into smaller exports each including no more than three tapes. Each internal export generates a single XML file. All files are output to the same directory.

Importing Tapes

Importing tapes to be used in restore operations is a two-step process. First, the metadata that describes the tape objects is imported using the importtapes command line utility. Once the metadata has been successfully loaded, the physical tapes can be inserted into the tape library using the **Insert** function in the Control GUI.

Note: Multiple simultaneous import operations are enabled, but not recommended.

Using the Import Command

To use the importtapes command you must first ensure that the exported XML metadata file and the .ffm files exist on the destination DIVA Core System. The files must exist in uncompressed form in the Manager's bin directory (*by default*). Also, the *Object Tape Group* must already exist on the target system before the import begins. This tape group does not necessarily have to be the same group assigned to the tape in the source system.

The three main ways that a tape object can be treated during the import process are as follows:

- Imported as a new object
- Skipped
- Added as an instance of an object already existing in the DIVA Core database

Import as New Object

Normally, when a tape object is imported by the utility it is imported as a new object. This can only occur when the *Object Name* and *Object Category* for the tape object does not exist in the target DIVA Core system. In the event of a naming conflict, the default behavior is to terminate the import operation without importing any tapes or objects.

When new objects are imported into the target DIVA Core system, the import function only looks at the XML and FFM files and does not read directly from the tape structure. SPM is also automatically notified and if the object matches any of the SPM filters, then SPM will initiate the required actions for the object.

Skip Object

Caution: You must be careful when skipping objects because the tape object that is skipped may or may not actually be the same as the object in the database. The tape object that had the naming conflict may in fact contain different content than the existing one in the DIVA Core database (*content that should be preserved*). If a tape is imported and then repacked, objects that were skipped will not be copied to the new tape and the old tape will be reclaimed. If all objects on a tape are skipped (*and the tape is made writable*), the tape will be marked for deletion and new objects will overwrite existing objects on the tape. If you write new objects to the tape after the last object on tape is skipped, that tape instance will immediately be overwritten.

A tape object can be skipped if the -skipIfNameExists flag is passed to the Import Utility. If there is another object already in the DIVA Core database that has the same *Object Name* and *Object Category* as a tape object being imported, and the -skipIfNameExists flag is set, the object is skipped. The object instance on the tape is not recorded in the DIVA Core database (*it is considered deleted by DIVA Core*), and processing continues with the next tape object in the import metadata.

Using the Import Date as the Archive Date

The TapeImport command line utility provides an additional command line switch named -useImportDateAsArchiveDate.

Using this switch during object import causes the date of the imported object to be used as the date of object archival in the system where it is being imported. The original archive date is not replaced in the XML export or on the original DIVA Core system, it is only replaced for the object on the imported system.

This feature supports tapes with spanned objects in the same way as regular tapes.

Add as an Instance

An object can be imported as an instance of another object if the -addAsInstanceIfNameExists flag is passed to the Import Utility. If there is another object already in the DIVA Core database that has the same *Object Name* and *Object Category* as a tape object being imported, and the -addAsInstanceIfNameExists flag is passed, an Import as an Instance is attempted.

First, the checksums for the tape object are compared to the checksums in the database object that matches it. If a match is produced (*for each object component*), the object is imported as an instance of the matching object. The ***Comments***, ***Archived Path Root***, ***Archive Date***, ***UUID***, ***Storage Plan***, ***Group***, and so on, of the imported object are lost and become that of the object already in the DIVA Core database.

A new Object instance ID is assigned every time the utility imports as an instance.

If the *Checksum Type* of the object components in the database does not match the *Checksum Type* in the imported object, or if one of the two objects has checksums that are missing, the tape object is not imported as an instance. This is considered a checksum mismatch and the import processing halts. However, if both the -skipIfNameExists flag and the -addAsInstanceIfNameExists flag are passed to the Import Utility (*and a tape object matches one that already exists in the DIVA Core database*), the utility first tries to import the object as an instance by comparing checksums. If this attempt fails the object is skipped and processing will continue.

Note: SPM is not notified when importing as an instance. If the object matches any of the SPM Filters then SPM will not initiate the required actions for the object.

Error Conditions

If the tape media is not recognized by the Manager an error will be generated specifying what occurred.

If the import process fails and Manager detects a database error, the import process will be terminated and any operations performed during the failed import will be rolled back and not saved in the system.

In the case where the checksum comparison failed (*or the checksum is not present*) for one or several objects, the entire import process will be stopped and the database transaction will be rolled back.

If the -skipIfNameExists flag is used, the checksum verification will still execute. However in this case an unverified (*mismatched*) object will be skipped instead of stopping the entire import process.

All errors are displayed on the screen and written to the log file. When using the -skipIfNameExists flag, you must check the screen messages and log file to determine whether all content intended to be imported was processed successfully. This option is not compatible with automated workflows since it may require operator intervention and decision.

Warnings and Limitations

Complex objects that are compared this way must have been archived in the same exact order to pass the checksum verification.

The Import Utility does not compare *UUID*, *Object ID*, *Archive Dates*, or *Site ID*. The ***Comments***, ***Archived Path Root***, ***Archive Date***, ***UUID***, ***Storage Plan***, ***Group***, and so on, of the imported object are not preserved when being added as an instance.

The utility does not enable the import of a set of tapes that contain an object with more than one instance on the tapes. An import metadata file having an object with more than one instance appearing within an exported tape set is not allowed. The export utility prevents this from happening.

Importing Encrypted Tapes

You only specify the destination group, and the folder containing the xml metadata file and the optional Keystore file to import encrypted tapes.

If a Keystore file is not present in the export folder, and you are attempting to import encrypted tapes, the tape encryption key for every tape must match the encryption key of the destination group.

If the Keystore file is present in the export folder, it will be opened using the password in the *Manager Configuration* panel of the new system, and therefore the password must match. If it does not match, you will be asked one time for the Keystore password before failing the import. During the import the encryption keys will be compared to the encryption key hash and salt to validate the keys. If any key is not valid, the import will terminate.

Example:

```
Importer default D:\workspace\Manager\bin\exported\2017-03-25--10.37.14
===========================================================================
DIVA Core Tape Importer
```

```
Copyright (c) 1999, 2017 Oracle and/or its affiliates. All rights reserved.
All rights reserved.
==========================================================================
The import completed successfully.
(Code 0)
```

Bulk Tape Import

After inserting all tapes into the new system in the default group, you can disable the ***Tape Protection Setting*** for all tapes. You change the setting on the *Inserted Protected Tapes* panel in DIVA Command, Select all tapes, click **Edit**, change the setting, and then click **OK**.

Import Tape Procedure

Importing of tapes is accomplished using a combination of the Windows command-line interface and the Control GUI. Inserting the tape is an optional part of the workflow but is necessary to access the objects on the tape. It is possible to run the importtape command line utility to enter the tape's metadata into the DIVA Core database and still keep the tape externalized. However, to access the objects on the tape, the tape must be inserted using the Insert Tape function.

The following procedure is used for importing tapes into DIVA Core:

1. Open a Windows command-line interface.
2. Copy the exported XML and FFM Files into the DIVA_HOME\Manager\bin folder.
3. Change to the DIVA_HOME\Manager\bin folder.
4. Run the importtape command using any of the following necessary command line options:

   help (-h)
   Displays help information.

   groupname
   The tape group to which imported tapes will belong. The group must already exist in the system.

   mfiledir
   The XML file containing exported tape metadata, or a folder that contains the files.

   -skipIfNameExists
   Skip import of objects with naming conflicts. The default behavior is that if the *Object Name* and *Object Category* already exist, the utility will terminate without importing the tape(*s*). Using this option in the command line will override the default.

   -addAsInstanceIfNameExists
   Attempt to add the tape object as an instance of an existing object in the DIVA Core database. The tape object must have the same *Object Name* and *Object Category*, components, and checksums as the object in the database.

   -useImportDateAsArchiveDate
   Changes the imported object's original archive date to the date of import on the destination system. This does not change the original archive date in the exported XML file or in the original system where the object was exported from, only on the system where the object was imported.

5. In the Control GUI, navigate to the **Home** tab, and then click the **Tapes** button to show the list of tapes identified in the system through the **Tapes** panel. Imported tapes can be left externalized, but to restore the objects on a tape it must be inserted into the library.
6. Navigate to the **Action** tab on the ribbon bar and click **Insert Tape** to open the Insert Tape dialog box.
7. If necessary, select the check box ***Require instances on tape(s)***. Otherwise leave it deselected if this is not required.
8. Select the appropriate ***Robot Manager Name*** using the menu list.

   An error is displayed if no Robot Manager is selected.
9. Select the appropriate ***CAP ID*** using the menu list.

   An error is displayed if no CAP ID is selected.
10. Use the slide control to select the priority value for the insert operation.
11. Restoration of the objects on the imported tapes is possible after the tapes are inserted.

Import Example

The tape with barcode number *000131* also contains objects that are spanned across the tape with a barcode of *000120*. When tape *000131* is exported, its exported XML File is named Tapeset-000131.xml. This XML file also includes the objects from tape *000120*, and both tapes *000131* and *000120* will be ejected from the library. After all objects from both tapes are exported to the XML file, all instances on each tape and references to the tapes themselves are removed from the DIVA Core database.

The XML file is then copied to the DIVA_HOME\Program\Manager\bin folder of the target DIVA Core system. The command importtapes MOVIES Tapeset-000131.xml results in the metadata for this tape being imported into the group *MOVIES*.

When the tape's metadata has been successfully imported to the database (*check the Control GUI **Current Requests** queue*), both of the tapes and their objects are considered externalized and can then both be entered into the library with the **Insert Tape** command.

Importing of WORM Media is supported by DIVA Core 7.4 and later. However, when you import DIVA Core 7.4 (*or later*) WORM media into a DIVA Core release earlier than DIVA Core 7.4, DIVA Core ignores the WORM flag (*set to false*) and logs it in the Manager log. The device will be seen in the Control GUI as a tape but not usable if finalized or no WORM drive is connected to the system.

8

DIVAmigrate Operations

This chapter describes DIVAmigrate operations and includes the following information:

Starting and Stopping the DIVAmigrate Service

You can start and stop the DIVAmigrate utility from the command-line, or the Windows Service Manager Console. The Windows Service short name displayed in the Windows Service Manager Console is DivaMgrt, and the full name displayed is DIVA Core Migrate - DIVAmigrate.

To install the DIVAmigrate utility, open a command-line interface, and navigate to the folder where the migrate.bat (*or migrate.sh in Linux*) file is located. Run the batch (*or script*) file with the appropriate parameters required to perform the desired tasks.

Caution: It is your responsibility to confirm that only one instance of the DIVAmigrate Service is running at any time. Running several instances of DIVAmigrate connected to same Manager leads to incorrect Migration Job processing, and may result in data loss.

The following is an example procedure to run the service on a Windows 2012 R2 platform.

1. Open a Windows command-line interface.
2. Change to the %DIVA_HOME%\Program\Migrate\bin folder.
3. Execute a command similar to migrate.bat install. This specific command installs the service.
4. Execute migrate.bat start to start the service. The service is now running.

The following is an example procedure to run the service on an Oracle Linux platform. *Linux commands, file names, and paths are case-sensitive*.

1. Open a command-line interface.
2. Navigate to /home/diva/DIVA/Program/Migrate/bin.
3. Execute migrate.sh.

The migrate.bat command-line syntax is migrate.bat {command} [option]. The available commands and options are as follows:

install
This command install the module as a Windows service.

uninstall
This command removes the module Windows service.

start
This command starts the service.

stop
This command stops the service.

restart
This command stops, and then subsequently starts the service.

version
This command displays the module release information, and then exits.

status
This command displays the current status of the module (*for example, running, stopped, and so on*).

help
This command displays command and option help information, and then exits.

-conf or -f
This parameter is optional and specifies a specific configuration file to load.

Migration Job Command Syntax

You can start DIVAmigrate Migration Jobs using either the Control GUI or the command-line interface. The command-line interface uses the client.bat batch file in Windows, or the client.sh script file in Linux. The Windows batch file is located in the %DIVA_HOME%\Program\Migrate\bin folder, and in the Linux script file is located in the /home/diva/DIVA/Program/Migrate/bin directory.

The client command expects all of the command line options to follow in the exact sequence specified by the syntax.

For example, the following command fails:

```
client -tape 1S0001 -buffer array1 -media group2 -count 1 -delete
```

The displayed error message is:

```
Error. Check parameters. Use client.bat -help for valid commands description.
```

This is because the -buffer option is out of sequence, and is expected to occur after the -media and -count options.

You can use the following syntax for the associated migration jobs:

```
client -tape {tape_barcode} [-media {media_group}] [-count {count}] [-buffer {array}] [-delete] [-excl {file_name}]
[-autosrc|recovery] [-start {date_time}] [-end {date_time}] [-priority {value}]

client -tapelist {file_name} [-media {media_group}] [-count {count}] [-buffer {array}] [-delete] [-excl {file_name}]
[-autosrc|recovery] [-start {date_time}] [-end {date_time}] [-priority {value}]
```

```
client -instlist {file_name} [-media {media_group}] [-count {count} [-buffer {array}] [-delete] [-excl {file_name}]
[-autosrc|recovery] [-start {date_time}] [-end {date_time}] [-priority {value}]

client -tapegroup {mediaName} [-media {media_group}] [-count {count} [-buffer {array}] [-delete] [-excl {file_name}]
[-autosrc|recovery] [-start {date_time}] [-end {date_time}] [-priority {value}]

client -diskarray {mediaName} [-media {media_group}] [-count {count} [-buffer {array}] [-delete] [-excl {file_name}]
[-autosrc|recovery] [-start {date_time}] [-end {date_time}] [-priority {value}]

client -{pause|stop|resume|cancel|delete|retry} {jobID}

client -list_running_jobs

client -job_details {jobID}

client -help
```

The command-line options in the previous statements must be in the correct sequence as shown, and are described as follows:

-tape {tape_barcode}
This option identifies the tape to migrate. This will migrate instances present on the tape for the specified barcode.

-tapelist {file_name}
This option identifies a file containing a list of tapes to migrate. This will migrate instances present on the tapes listed in the file. The file is a flat text file (*DOS or UNIX format*). The format of each line is barcode. This option ignores empty lines and lines beginning with #.

-instlist {file_name}
This option identifies a file containing a list of instances to migrate. The file is a flat text pipe character separated file (*DOS or UNIX format*). The format of each line is objectname|category|instanceid. This option ignores empty lines and lines beginning with #.

-tapegroup {mediaName}
This option migrates all object instances in the entire tape group. The mediaName is the tape group name.

-diskarray {mediaName}
This option migrates all object instances in the entire disk array. The mediaName is the disk array name.

-media {media_group}
This option identifies the media where objects are being migrated. Specifying a different media group other than the Source Server media enables various migration operations. For example, migrating data from one media to another. When this option is not specified, the destination media group is the Source Server media group of the tape or instance being processed, and corresponds to the COPY_TO_SOURCE migration strategy. If multiple destination media are specified (*by using multiple -media options*), you must provide the same amount of -count options in the same order (*or none to use the default*). *This option is mandatory with the -instlist and -tapelist options*.

-count {count}
This option identifies the target instance count for the objects in the destination media. The default value is 1. If multiple destination media are specified (*by using multiple -media options*), you must provide the same amount of -count options in the same order (*or none to use the*

default). When specifying a COPY_TO_SOURCE migration type (*where source and target media are the same*), -count 2 is required rather than the default of -count 1. This specifies that there will be two instances on the target medium.

-buffer {array}
This option identifies the disk array to use as a buffer.

-delete
This option deletes the Source Server instances after they are processed. This option represents the MOVE migration strategy.

-excl {file_name}
This option identifies a file containing a list of instances to ignore. The file is a flat text file (*DOS or UNIX format*). The format of each line is object|category|instanceId. The object is the object name to exclude, and category is the object category. You can use an asterisk to match all object names or all categories. These are mutually exclusive. The instanceId is the ID of the instance. Blank spaces in object, category, and instanceId are acceptable. Objects and categories containing a pipe operator within the name are permissible, but the character must be escaped using the backslash character. For example, you must write object|one as object\|one.

-autosrc
This option enables using alternate Source Servers. The Migration Service can search whether the source tape instance exists on an array and can use it for migration, rather then using the tape instance.

-recovery
This option enables recovery mode.

-start {date_time}
This option specifies the scheduled start time for a job in the format dd-MM-YYYY-HH:mm.

-end {date_time}
This option specifies the scheduled end time for a job in the format dd-MM-YYYY-HH:mm. The job enters the **PAUSE** state if the job is not completed before the scheduled end time.

-priority {value}
This option specifies the migration job request priority. This value must be between 1 and 100. The default value is 20.

-[command] {jobID}
This option sends the command used for the specific job. Commands include pause, stop, resume, cancel, delete, and retry.

-list_running_jobs
This option displays a list of unfinished jobs, including the jobID and status.

-job_details {jobID}
This option returns the status and progress for a given jobID.

-help
This option displays the help text.

Using the DIVAmigrate GUI

The Control GUI contains the *Migrate* panel and *Migration Wizard*. You use the Migration Wizard to create new migration jobs. You view the status and results of completed and currently running Migration Jobs on the *Migration* panel. You can also pause, stop, resume, delete, cancel or rerun migration jobs from the *Migration* panel.

Migration Job Events

The migration service includes events for jobs. All job events are displayed under the **Job Events** tab in Job Properties dialog box. Events are loaded in descending order according to time and event id (*by default*). 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 informational messages. Click the **Refresh** button to refresh the entire *Job Properties* dialog box.

Using the DIVAmigrate Migration Wizard

The Migration Wizard is a typical Windows wizard with a series of dialog boxes. Each dialog box presents a set of choices for migration job parameters and, based on your selections, displays the next screen until all required parameters are configured.

The **Create Migrate Job** button is located under the **Migrate Actions** button on the **Action** tab. You click the button to start the Migration Wizard.

The DIVAmigrate Wizard is only available to users with Administrator privileges in the Control GUI. Use the following procedure to perform a migration using the wizard:

1. Navigate to, and then click, the **Create Migration Job** button in the Control GUI.
2. Select the ***Migration Strategy*** and click **Next**.

Note: If you selected the **Copy to Source *Migration Strategy***, then *Step 3* is skipped because only the ***Media Option*** is available for this strategy. Continue the procedure with *Step 4*.

3. Select the ***Migration Source Type*** and click **Next**.
4. Use the ***Migration Source*** menu to select the Source Server, and then click **Next**.
5. Use the ***Destination Media*** menu to select the destination.
6. Enter the number of instances to migrate, and then click **Next**.
7. Select **Yes** to add more migration destinations, or **No** to proceed. If you select **Yes**, the previous screen is displayed again so you can select additional destinations. This process repeats until you select **No**.
8. Click **Next** when you have finished adding destinations.
9. Select, or enter, the value in each field appropriate for this migration job on the Migration Options screen, and then click **Next**.
10. The final wizard screen now displays with your selections made during the process. On this screen you can either click **Finish** to schedule the job, **Cancel** to exit the Wizard (*no job will be created*), or **Previous** to go back and make any necessary changes.
11. Confirm your selections, and then click the appropriate button to either finish, modify, or cancel the migration job.

Using the DIVAmigrate Panel

The *DIVAmigrate* panel is located in the Control GUI and displays information about migration jobs in a table containing the following columns. Some columns are filterable as noted.

- **ID** (*this value is a number*)
- **Status** (*filterable*)
- **Type** (*filterable and the value can be Copy, Move, or Copy to Source*)
- **Source Server Type**
- **Destination Type**
- **Started At** (*filterable*)
- **Finished At** (*filterable*)
- **Progress**
- **Instances to Migrate**
- **Data to Transfer** (*Gb*)

A single row selection model is used for the table, and the context dialog is available for each row with following options. See Migration Job Actions for detailed information.

- **Properties** (*opens selected migration job properties dialog box*)
- **Cancel**
- **Stop**
- **Pause**
- **Resume**
- **Rerun**
- **Delete**

The Request Properties dialog box shows the status for each object affected by the migration job, and the *jobID* of the last request executed for that object. You click the *Request ID* to open the Request Properties dialog box. The *Object Details* table in the Migration Job Properties dialog box displays 1000 objects at a time and supports pagination.

Migration Job Functions and Parameters

This sections describes the migration job functions and parameters.

Caution: DIVAmigrate does not allow any changes to the migration job parameters and options after a migration job is submitted.

Migration Job Definitions

You create migration jobs using DIVAmigrate, and then DIVAmigrate attempts to run them. You must configure each of the following mandatory parameters for each migration:

Migration Source

This parameter describes the set of instances that must be migrated.

Migration Destination

This parameter specifies the location where migrated instance are to be placed.

Migration Strategy

This parameter specifies the type of migration. That is, **Copy**, **Move**, or **Copy to Source**.

Migration Options

These are additional parameters and options available for migration jobs.

Migration Job Actions

You can perform the following actions on migration jobs:

Create New Job

This selection creates a new migration job.

Cancel Job

This selection stops the specified job without the possibility of resuming. The job is assigned the **Cancelled** status.

Rerun Job

This selection creates a new migration job using the same parameters as the original job.

Pause Job

This selection saves the current state of the migration and pauses job execution. After resuming, the job continues execution from the point where it was paused without recollecting data.

Stop Job

This selection stops the job, but maintains the ability to resume later. After resuming, the job's internal state is completely reinitialized, and all migration data is recollected.

Delete Job

This selection deletes the job from the database. If the job was running, or pending, it is stopped first and then deleted. This works for all job states (***Pending, Running, Completed,*** *and so on*).

Resume Job

This selection resumes the job's execution for a job in a **Stopped** or **Paused** status.

Migration Job Status

Based on the migration job lifecycle, and possible user actions, each migration job has one of the following statuses:

Submitted

This status indicates you configured the migration job, but DIVAmigrate did not start working on it yet.

Pending

This status indicates that DIVAmigrate found a new migration job and started processing the job.

Initializing

This status indicates that DIVAmigrate is collecting all necessary information required to start the migration job.

Copying Content

This status indicates that DIVAmigrate is creating copies of instances selected for migration.

Cleaning Up

When the **Move** migration strategy is selected, DIVAmigrate deletes already migrated instances from the migration source. If a disk buffer is used, DIVAmigrate removes the instances from the buffer.

Finalizing

This status indicates that DIVAmigrate is checking the migration results, and updating the migration job status and other job information.

Completed

This status indicates that the migration job was successfully completed.

Partially Completed

This status indicates that the migration job was completed, but some objects were not migrated.

Cancelling

This status indicates that the command to cancel the migration job was received, but has not yet been processed by DIVAmigrate.

Cancelled

This status indicates that you canceled the migration job.

Aborted

This status indicates that the migration job was not completed due to an error.

Pausing

This status indicates that the command to pause the migration job was received, but has not yet been processed by DIVAmigrate.

Paused

This status indicates that the migration job was paused by DIVAmigrate.

Stopping

This status indicates that the command to stop the migration job was received, but has not yet been processed by DIVAmigrate.

Stopped

This status indicates that the migration job was stopped by DIVAmigrate.

Resuming

This status indicates that the command to resume the migration job was received from the client (*GUI*), but has not yet been processed by DIVAmigrate.

Deleting

This status indicates that you submitted the command to delete the migration job entirely.

Migration Job Parameters

This section describes (*in detail*) each of the migration job parameters.

Migration Source

This ***Migration Source*** parameter defines the set of instances for migration. You can provide the ***Migration Source*** in any of following ways:

- Barcode for a single tape

- List of Tapes
- Single Tape Group (*mediaName*)
- Single Disk Array (*mediaName*)
- List of Instances (*object name, object category, and instance ID*)

You must provide valid values for the ***Migration Source*** when using *List of Tapes* or *List of Instances*. DIVAmigrate will not try to validate List of Tapes, List of Instances and the Excluded Instances List before starting the migration job.

The asterisk mask filter is not supported when you use List of Tapes or List of Instances as a ***Migration Source***. The asterisk mask filter is only supported for the Excluded Instance List.

You have the option to provide the Tape List and Instances List in a file. The following rules apply to the file:

- File names can have any name and extension.
- For Tape Lists, each line in the file should contain only one barcode.
- For Instances Lists, each line in the file must contain the object name, object category and instance ID. They must be exactly in this sequence, and you must use the pipe operator as the separator.

  For example, objectname|category|instanceid.

Note: Character escaping is supported if the pipe operator is present in the object name or category. For example, you must write object|one as object\|one.

Migration Destination

This ***Migration Destination*** parameter defines the target media where objects are migrated. The valid ***Migration Destinations*** are **Single Media**, **Multiple Media**, and the **Same as Source** as the ***Migration Source*** (*for the **Copy to Source** migration strategy*).

The following rules apply to the number of new instances being created on the ***Migration Destination***:

- DIVAmigrate checks how many instances of each object being migrated already exist on the destination, and counts already existing instances as already migrated.

  For example, one instance for *Object-A* already exists on the destination media called default. You set the number of instances on the destination media (*default*) to 1. The migration job performs no additional copy of *Object-A* to the destination media (*default*) because one instance already existed before the migration job started.
- If you select **Multiple Media** for the ***Migration Destination***, you must provide the number of instances that must be present for each media separately.
- If you create a job with multiple destinations, using a combination of both tape groups and disk arrays, the migration service always migrates to the disk destination first.
- If the job has multiple media destinations, and one of the destinations is a disk array, and the source is a single tape, tape group, or tape instance, DIVAmigrate always uses instances migrated to the disk destination as the Source Server for migration to any tape destinations. If the alternative instance's migration fails, the original Source Server instance is used for migration.

Migration Strategy

The ***Migration Strategy*** parameter defines the type of migration being performed. The following ***Migration Strategy*** options are currently implemented:

Copy

This strategy migrates the configured number of instances from the ***Migration Source*** to the ***Migration Destination***, but does not remove original instances from the Source Server.

Copy to Source

This strategy is the same as the **Copy Strategy**, but the ***Migration Source*** and ***Migration Destination*** are the same. This strategy is applicable only for a **Single Media** source.

Move

This strategy migrates the configured number of instances from the Source Server to the Destination Server, and deletes original instances from the source.

Migration Options

The following Migration Options are currently implemented:

Requests Priority

This parameter identifies the priority of requests sent to the Manager by the DIVAmigrate Service during the migration job. *This is a mandatory parameter*.

Excluded Instances List

This parameter is a list of instances you provide. The instances in this list are excluded from the migration. You have the option to input the list as a file with values separated by the pipe operator, or you can enter them manually. Both the file and manual input process use the following similar conventions:

- The object name and object category support filtering using an asterisk.
- The instance ID can be a number or an asterisk. An asterisk indicates any and all instances.

For example, `objectname|category|instanceid`. *This is an optional parameter*.

Alternative Instance Sources

If you enable this option, DIVAmigrate performs the following processing for each instance it attempts to migrate:

- If the instance is on tape, DIVAmigrate checks for a disk instance of the same object anywhere in the system and uses it instead.
- Alternative sources are selected during the initialization stage of a migration job, and are not updated during the migration job processing.
- The migration tool always migrates the disk instances available on the alternative disk first as part of the migration plan.

DIVAmigrate always considers all disk instances to be online and does not perform an availability check for them. If the disk where the instances are located is broken, the migration of the instance fail and the error is logged.

The original instance is always used if the migration of an alternative instance fails. *This is an optional parameter and is turned off by default*.

Recovery Mode

Recovery Mode is an extension to the ***Alternative Instance Sources*** option, which allows searching for any online instance (*both disk and tape*) anywhere in the DIVA Core system, and enables implementation of recovery scenarios for damaged offline tapes.

When this option is on, the ***Alternative Instance Source*** option is automatically also on. If the migration operation fails during the migration of an alternative recovery instance, the original Source Server instance is used for migration.

This is an optional parameter and is turned off by default.

Disk Buffer Name

You can specify the disk array name to use as the migration's ***Disk Buffer Name***. Currently, you can only assign one disk array as a disk buffer. Tape objects are initially migrated to the disk buffer from ***Migration Source***. After being migrated to the disk buffer, they will be migrated from disk buffer to the identified ***Migration Destination***.

You configure the maximum percentage of space available for migration on the disk arrays in DIVA Command. During the migration process, the maximum used space on the disk buffer is periodically compared to the value of this parameter. If the percentage of used space is greater than, or equal to, the configured value, then the migration job attempts to migrate objects from the disk buffer to the destination. The disk buffer is then cleaned up, and the job continues with the remaining objects being migrated. If, after disk buffer clean up is performed, there is still not enough free space the job will be paused.

The following rules are applied when using the disk buffer:

- The Migration Service does not check the disk buffer availability status before submitting a request.
- If the disk buffer runs out of space during migration job processing, all currently cached objects are migrated from the disk buffer to the ***Migration Destination***, and then cleaned from the disk buffer. After that, the Migration Job continues processing as usual.
- If the ***Alternative Sources*** option is on, and for a tape instance there is alternative disk instance, the instance is not copied to the disk buffer.
- Instances cached to the disk buffer are deleted from it as part of the same migration job.
- DIVAmigrate checks if the disk buffer percentage of used space is less than the configured value for each disk array. If it is more than the configure value, then the migration job is paused.
- The disk buffer is not used for instances that are migrated from a disk Source Server, an alternative disk Source Server, or to a disk Destination Server.

This is an optional parameter.

Scheduled Start Time and *Scheduled End Time*

You can schedule migration jobs by providing a ***Scheduled Start Time***, a ***Scheduled End Time***. You can provide none, both, or just one of them.

For a ***Scheduled Start Time***, DIVAmigrate guarantees not to start the migration job until this time is reached. DIVAmigrate attempts to start the job as soon as the scheduled time is reached if it has enough available resources.

For a ***Scheduled End Time***, DIVAmigrate guarantees to stop the job processing after this time is reached. A stopped migration job receives the **Paused** status. The ***Scheduled End Time*** parameter is removed from the job after it resumes.

The DIVAmigrate Service validates your input for new migration jobs in the GUI when you create a job, and in the Service when the job is initialized. No job parameter validation is performed for jobs submitted through the command line.

*These are both optional parameter*s.

Basic Migration Jobs

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 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 format.

Copying Data to another Group or Array

The following scenarios describe different possible outcomes when you copy data to another group or array using the DIVAmigrate Utility.

Scenario 1

A series of objects (*Object-A, Object-B, and Object-C*) are in the *Main Tape Group* and must be duplicated to the *Backup Tape Group*. The following table identifies the initial state and location of the tapes:

Main Tape Group	Backup Tape Group
Object-A Object-B Object-C	

You submit a copy job to create a copy of *Object-A*, *Object-B*, and *Object-C* to the *Backup Tape Group*. This command results in *Object-A*, *Object-B*, and *Object-C* having instances in both tape groups as follows:

Main Tape Group	Backup Tape Group
Object-A Object-B Object-C	Object-A Object-B Object-C

Scenario 2

When the initial state includes one or more objects in the *Main Tape Group* already existing in the *Backup Tape Group*, the initial state is as shown in the following table:

Main Tape Group	Backup Tape Group
Object-A Object-B Object-C	Object-A

You submit a copy job to create a copy of *Object-A*, *Object-B*, and *Object-C* to the *Backup Tape Group*. This command results in *Object-A*, *Object-B*, and *Object-C* having instances in both tape groups as follows:

Main Tape Group	Backup Tape Group
Object-A	Object-A
Object-B	Object-B
Object-C	Object-C

The result is one instance of each object in *Backup Tape Group*.

There are many possible reasons for *Object-A* already being in the *Backup Tape Group*, including (*but not limited to*) the following:

- A user manually copied the object to the *Backup Tape Group*.
- At one time, a user possibly configured the Storage Plan Manager to make copies to the *Backup Tape Group*, and then either disabled the corresponding slot or modified the configuration.
- The DIVAmigrate Utility was requested to duplicate objects, and then the process was interrupted before it completed.

In production environments, it is not generally known how many object instances currently exist in the target group (*possibly none*). However, it is typically known exactly how many are desired. In the previous example, the desired result is to have one instance in the *Backup Tape Group* for each *Main Tape Group* object after the duplication run. The DIVAmigrate Utility was designed to perform in this manner. The utility expects you to specify the desired number of instances in the target group or array. Depending on the actual situation, DIVAmigrate only performs the copy operations required to achieve the desired result.

Moving Data to another Group or Array

You will eventually encounter scenarios where moving the data is required, rather than replicating it. For example, this is useful when migrating data to a new technology (*for example, from LTO-3 to LTO-8*), and it is not necessary to keep the old tape instances.

In the following scenario, a new *Tape Group Main-LTO-8* was created and assigned a Tape Set ID populated with LTO-8 tapes. The following table displays the initial state:

Main Tape Group	Tape Group Main-LTO-8
Object-A	
Object-B	
Object-C	

You use the following command to instruct DIVAmigrate to perform the job correctly:

```
client {object list} -media Main-LTO-8 -count 1 -delete
```

This command includes the -delete option, which tells DIVAmigrate to delete the Source Server instances after copying them. In the example, the instances of *Object-A*, *Object-B*, and *Object-C* located in *Main Tape Group* are removed.

DIVAmigrate uses the following process for each object to complete this job:

1. Copy the object to the destination *Tape Group Main-LTO-8*.
2. Run DeleteInstance on *Main Tape Group*.
3. Mark the process complete, and repeat the process for the next object in the list.

This sequence is repeated until all objects have been moved. The final state is as follows:

Main Tape Group	Tape Group Main-LTO-8
	Object-A Object-B Object-C

The instances were moved to the new tape group. However, they are essentially different instances of the objects because the original instances were actually copied to the new tape group, and then deleted from the Source Server tape group.

Each time DIVA Core copies an object, a new instance of that object is created, and it is identified by a new Instance ID. In this scenario, the instance of *Object-A* located in the new *Tape Group Main-LTO-8* has a different Instance ID than the instance that was located in *Main Tape Group*. Because the new instance is a copy of the same object, the data is the same, and the data was actually moved.

The following table indicates the possible Instance ID for each object. The instances deleted after the run are shown in the *Main Tape Group*, and marked deleted in the table. The *Main Tape Group* still exists, but the instances in it no longer exist because they were deleted.

Main Tape Group	Tape Group Main-LTO-8
Object-A, Instance ID 1 (*deleted*)	Object-A, Instance ID 2
Object-B, Instance ID 1 (*deleted*)	Object-B, Instance ID 2
Object-C, Instance ID 1 (*deleted*)	Object-C, Instance ID 2

Copying and Migrating Data to the same Group or Array

Sometimes moving or replicating data to the same group or array as the one containing the Source Server instances is required. For example, if you are performing a tape technology migration, and you want to use the existing tape group name for instances stored on the new technology tapes. The old and new technology tapes are placed in the same SetID and the same group will contain both tape types. Telestream recommends setting the old type tapes to *Not Writable* (or *Protected*) to force DIVA Core to write new data to the new technology tape type.

You may find it efficient to create a group that only contains the desired new technology tapes and use that as the ***Destination Media***.

The following is an example of migrating objects from LTO-3 to LTO-8 tapes in the same *Main Tape Group*. The table shows the initial state of the objects.

LTO-3 Tapes	LTO-8 Tapes
Object-A, Instance ID 1 Object-B, Instance ID 1 Object-C, Instance ID 1	

You use the following command for replicating the data:

```
client -tapelist tapelist.txt -media Main -count 2
```

- The tapelist.txt parameter is a file listing the barcodes of the tapes being migrated. If you are performing a technology migration, the file will contain all LTO-3 barcodes of *Main Tape Group*. Telestream Support can provide you with tools to generate the listings automatically.

- The -media Main parameter tells DIVAmigrate to copy the objects to *Main Tape Group*, which is the same group as the source data.
- The -count 2 parameter is important because DIVAmigrate is replicating to the same ***Destination Group*** from the same ***Source Group***. Therefore, two instances of each object after the job completes are present instead of one.

If you specify -count 1 instead of -count 2, DIVAmigrate verifies that one instance exists for each object. Because this is already the case, DIVAmigrate considers the job complete and does nothing. This is a good way to perform a dry run with the DIVAmigrate Utility for a quick sanity check.

This is the final outcome after the job completes:

LTO-3 Tapes	LTO-8 Tapes
Object-A, Instance ID 1 Object-B, Instance ID 1 Object-C, Instance ID 1	Object-A, Instance ID 2 Object-B, Instance ID 2 Object-C, Instance ID 2

You can use the following command to move the data instead of replicating it:

```
client -tapelist tapelist.txt -media Main -count 1 -delete
```

- The -delete parameter tells DIVAmigrate to delete the original instances after they have been copied.
- The -count 1 parameter tells DIVAmigrate that there must be only one instance of each object after the job completes.

Executing the previous command results in the following final state:

LTO-3 Tapes	LTO-8 Tapes
	Object-A, Instance ID 2 Object-B, Instance ID 2 Object-C, Instance ID 2

Because DIVAmigrate can skip unnecessary tasks, you can use the previous two example commands one after the other with the same results. The first command replicates the data, and the second command only performs the deletes. DIVAmigrate recognizes that the copies are already completed and skips that step in the second command.

You can use this two-step migration if you do not initially have faith in your new technology tape drives. If you prefer to keep the old instances temporarily, for data safety, use the first (*replicate*) command as a first step. You only run the first command until you have been using the new technology tapes and drives for a while to save a copy of the old objects. When you are satisfied with their functionality and stability, you can then use the second (*move*) command to delete the old instances.

Stopping and Resuming Jobs

The DIVAmigrate Utility only performs the steps necessary to reach the intended goal, and nothing more. Any time a migration job is started, DIVAmigrate examines the DIVA Core catalog and calculates which copy and delete operations remain incomplete. It does not matter to DIVAmigrate if the job was already partially completed during a previous run, it will always adapt to the current state without the need to store any progress in a status or log file.

For this reason, you can freely interrupt a DIVAmigrate job at any time and resume it later. There are no specific actions to complete before resuming an interrupted job.

Advanced Migration Jobs

This section describes advanced migration operations.

Speeding up Tape to Tape Migration using a Disk Buffer

The DIVAmigrate Utility processes objects sequentially. For example, if an object is copied from a *Tape-1* to a *Tape-2*, a Copy request is sent to the Manager. The Manager allocates some *Cache Disk* space to store the object and proceeds in two sequential steps:

1. Read the object from *Tape-1* and write it to the *Cache Disk*.
2. Read the object from the *Cache Disk* and write it to *Tape-2*.

DIVAmigrate also processes series of objects. For example, migrating a tape volume to another tape group. DIVAmigrate uses the following sequential procedure:

1. Read *Object-A* from *Tape-1* and write it to the *Cache Disk*.
2. Read Object-A from the Cache Disk and write it to *Tape-2*.
3. Read Object-B from *Tape-1* and write it to the Cache Disk.
4. Read Object-B from the Cache Disk and write it to *Tape-2*.

This process continues until all objects on the tape volume have been processed.

You can possibly write the data to more than one tape in the ***Destination Tape Group*** based on the activity and Manager workflow. However, for this example consider that all objects go to a single *Tape-2*.

During Step 2 in the previous sequence, the Source Server drive is idle and you can assign it to other requests. So *Tape-1* might dismount and another tape mounts. If this occurs, you must mount *Tape-1* and position it again in Step 3. This will probably require dismounting a tape from the Source Server drive before you remount *Tape-1*. This causes a substantial delay in operations. While Step 3 proceeds, *Tape-2* might get dismounted because the destination drive is now idle. Therefore, you may need to be mounted and positioned again in *Step-4*.

A large migration using tape to tape copies can result in tremendous dismount, mount, and positioning overhead. Telestream recommends that heavy migration operations are conducted on dedicated Datahubs with dedicated tape drives.

Even with dedicated tape drives and no mount and dismount overhead, you might observed that one drive is idle while the other is working. For example, in the previous sample sequence, the target drive is idle in Step 1 and Step 3, and the Source Server drive is idle in Step 2 and Step 4.

To suppress drive idle time, you can use a disk array as a buffer for the migration job. When a *Disk Buffer* is specified, DIVAmigrate first enters a *Read Phase*. During this phase, it continuously reads the Source Server tape and stacks as many objects as space allows in the designated array. It then switches to the *Write Phase*, where it continuously writes objects from the disk to the destination tape.

Caution: You must choose the array used as a disk buffer carefully. There are two very important rules to consider:

- Make sure that the array is not managed by the Storage Plan Manager. When SPM manages an array, the data DIVAmigrate stores there might be replicated to tape and deleted automatically. This hinders DIVAmigrate operations, and ends up with undesired tape instances, resulting in a waste of tape storage space.
- Make sure the array you use is not a repository containing resident instances of the objects being migrated. If it is, DIVAmigrate considers those instances to be *buffered instances* from a previous job, and they are deleted after the job completes.

The obvious benefits are that only one drive is used in each phase, and it is used continuously. There is still a possibility that the drive will be dismounted between two Copy requests and assigned to a more prioritized request from the Manager queue. However, this will not happen if the drives and Datahubs are dedicated to the migration.

There are recognizable similarities between this workflow and the way the Repack workflow functions. The difference is that DIVAmigrate creates standard object instances on a disk array, and must explicitly delete them after the job. Repack workflows create temporary instances on a cache disk that are cleaned by the Manager after the request. You must restart a terminated migration job using the *exact same command line*, and specifically, the *exact same buffer array* you used in the original job. If you change arrays between two runs, the second run overlooks the buffered instances from the first run, and recreates new instances in the new array resulting in:

- Wasted time re-creating disk instances that were already created on the previous array.
- Wasted space in the previous array because the instances created there are no longer managed by a migration job, and are never deleted.

The following is a sample migration command using a *Disk Buffer*:

```
client -tape 000001 -media Main-LTO-8 -count 1 -buffer NAS001
```

Sample Scenario

This scenario is an example using the previous command line where *Tape 000001* is an LTO-3 tape from the *Main Group*, that must be migrated to *Tape Group Main-LTO-8* using the *NAS001* array as a buffer.

Tape 000001 contains *Object-A* and *Object-B*. DIVAmigrate performs the following action sequence, assuming the objects do not have instances in the new tape group:

1. Copy Object-A from *Tape 000001* to *Array NAS001*.
2. Copy Object-B from *Tape 000001* to *Array NAS001*.
3. Copy Object-A from *Array NAS001* to a tape in *Tape Group Main-LTO-8*.
4. Delete the Object-A instance on *Array NAS001*.
5. Copy Object-B from *Array NAS001* to a tape in *Tape Group Main-LTO-8*.
6. Delete the Object-B instance on *Array NAS001*.

According to this workflow, instances are copied to the buffer in a single stream. This way the Source Server drive is kept busy. Each buffered instance is discarded as soon as it is no longer needed. That is, after the target copies of the related object are complete.

To delete instances from the Source Server tape after migration, you add the -delete option as follows:

```
client -tape 000001 -media Main-LTO-8 -count 1 -buffer NAS001 -delete
```

This is the corresponding action sequence:

1. Copy Object-A from Tape 000001 to Array NAS001.
2. Copy Object-B from Tape 000001 to Array NAS001.
3. Copy Object-A from Array NAS001 to a tape in Tape Group Main-LTO-8.
4. Delete the Object-A instance on Array NAS001.
5. Delete the Object-A instance on Tape 000001.
6. Copy Object-B from Array NAS001 to a tape in Tape Group Main-LTO-8.
7. Delete the Object-B instance on Array NAS001.
8. Delete the Object-B instance on Tape 000001.

The workflow is the same as the previous workflow, except that this time, after deleting the buffered instance, the Source Server instance is also deleted. If no errors were encountered, *Tape 000001* is empty after the migration job.

Creating Multiple Instances in the Destination Group or Array

DIVAmigrate can create any number of instances in the target group or array using the -count option. DIVAmigrate performs as many copies as necessary to reach the specified count.

DIVA Core never stores two instances of the same object on the same volume or disk. Therefore, if a count greater than 1 is specified, the ***Destination Tape Group*** or ***Destination Disk Array*** must have more than one tape volume or disk identified within it.

If a *Disk Buffer* is used (*-buffer option*), the instances are buffered only once regardless of how many instances are being created in the ***Destination Tape Group*** or ***Destination Disk Array***.

Assuming that *Backup* is a tape group, *Tape 000001* contains *Object-A* and *Object-B*, and these objects do not have instances in *Tape Group Backup*, the following command translates into the following sequence:

```
client -tape 000001 -buffer NAS001 -media Backup -count 2
```

1. Copy Object-A from Tape 000001 to *Array NAS001*.
2. Copy Object-B from Tape 000001 to *Array NAS001*.
3. Copy Object-A from *Array NAS001* to a tape in Tape Group Backup.
4. Copy Object-B from *Array NAS001* to a tape in Tape Group Backup.
5. Copy Object-A from *Array NAS001* to another tape in Tape Group Backup.
6. Delete the Object-A instance on *Array NAS001*.
7. Copy Object-B from *Array NAS001* to another tape in Tape Group Backup.
8. Delete the Object-B instance on *Array NAS001*.

The two instances of each object in the backup group were created separately. DIVAmigrate performs the first copy of Object-A and Object-B, then the second copy, instead of creating two copies of Object-A sequentially, then two copies of Object-B. This is because there is only one tape drive to use. If the same object is copied twice to the ***Destination Tape Group***, the Manager must use a different tape for each instance because DIVA Core does not allow two instances of the same object to be stored on the same volume. This causes a dismount and mount operation.

By performing the first copy of Object-A and Object-B sequentially, you can keep the same tape on the drive for both objects. You must only change the tapes when entering the second copy phase. DIVAmigrate refers to this as a *Migration Step* in its logs. There are two migration steps, a first copy to the *Backup Tape Group*, and then a second copy to the same group. DIVAmigrate always attempts to process as many objects as possible sequentially within a particular migration step. This minimizes the mount and dismount overhead.

The Migrate Utility does not remove extraneous instances in the ***Destination Tape Group*** or ***Destination Disk Array***. If one instance of each object is requested in the ***Destination Tape Group*** or ***Destination Disk Array***, and some objects already have two or more existing instances, DIVAmigrate considers the job complete for these objects because there is at least one instance. It will not delete some existing instances to reduce the count back to one.

Migrating to Multiple Destination Groups or Arrays

You can use more than one -media option for DIVAmigrate to copy data to more than one group or array. Each -media option must be accompanied by a -count option, and the default count is 1. For example:

```
client -tape 000001 -media Main -count 1 -media Backup -count 2 -delete
```

If a *Disk Buffer* is used (*-buffer option*) the instances are buffered only once, regardless of how many ***Destination Tape Groups*** or ***Destination Disk Arrays*** are specified.

Assuming *Backup* and *Ext* are tape groups, *Tape 000001* contains *Object-A* and *Object-B*, and these objects do not have instances in any ***Destination Group***, the following command translates into the following sequence:

```
client -tape 000001 -media Backup -count 2 -media Ext -count 1 -buffer NAS001
```

1. Copy Object-A from Tape 000001 to *Array NAS001*.
2. Copy Object-B from Tape 000001 to *Array NAS001*.
3. Copy Object-A from *Array NAS001* to a tape in *Tape Group Backup*.
4. Copy Object-B from *Array NAS001* to a tape in *Tape Group Backup*.
5. Copy Object-A from *Array NAS001* to another tape in *Tape Group Backup*.
6. Copy Object-B from *Array NAS001* to another tape in *Tape Group Backup*.
7. Copy Object-A from *Array NAS001* to a tape in *Tape Group Ext*.
8. Delete the Object-A instance on *Array NAS001*.
9. Copy Object-B from *Array NAS001* to a tape in *Tape Group Ext*.
10. Delete the Object-B instance on *Array NAS001*.

As explained in the previous section, DIVAmigrate divides the sequence in several individual migration steps. There are three in the previous example:

- A first copy to Backup.
- A second copy to Backup.
- A copy to *Ext*.

For each migration step, DIVAmigrate processes all objects in a row to minimize the amount of mount and dismount overhead. If this is not done, it would require three tape changes per object:

- One for the second copy to the same group.
- One for the copy to a different group.
- One for the switch back to the initial group for the next object.

Default Destination Instance Count

Omitting the destination instance count option (-*count*) results in DIVAmigrate using a default value of 1. The following examples assume migration of the data from *Tape 000001* belonging to *Tape Group Main*:

client -tape 000001 -media Backup
The destination count is set to 1 by default. This command tells DIVAmigrate to create an instance of each object from Tape 000001 in *Tape Group Backup*; replicating the data on Tape 000001.

client -tape 000001 -media Main
This command creates an instance of each object from Tape 000001 in *Tape Group Main*. However, this is the same group as the source tape so there is already an instance of each object in that group (*the one on Tape 000001*). DIVAmigrate considers the job already complete and does nothing more. To force the duplication of Tape 000001 to the same group, the -count 2 option must be added.

client -tape 000001 -media Main -delete
This command creates an instance of each object from Tape 000001 in *Tape Group Main*. This is the same group as the source tape, but this time DIVAmigrate is told to delete the source instances. Therefore, DIVAmigrate cannot include the source instance in the final count as it did in the previous example. Instead, DIVAmigrate creates an additional instance of each object in *Tape Group Main* before deleting the original instance on Tape 000001. The result is similar to a Repack operation.

Repacking Tapes

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.

To repack a suspected corrupt tape, using DIVAmigrate may be more effective than executing a Repack operation because it does not halt on errors. For example, the following command will mimic a Repack of a *Tape 000001* belonging to *Tape Group Main*:

```
client -tape 000001 -media Main -count 1 -delete -recovery
```

The -delete option tells DIVAmigrate to move the data. That is, copy it to the *Destination Group* and then delete the original instance.

The -count 1 option tells DIVAmigrate that after the migration completes, there should only be one instance of each object in *Tape Group Main* remaining.

The -recovery option causes DIVAmigrate to look for, and use if available, any alternative online instances either on disk or tape. You use this option if the source tape is known to be corrupt. If no alternative is found, DIVAmigrate still attempts to use the instance from the source tape.

The DIVAmigrate Utility copies each Source Server instance to the same group, and then deletes the Source Server instance. This is similar to a Repack operation.

There are, however, a few differences with Repack operations:

- Repack is a single request in the Manager queue. An equivalent DIVAmigrate job produces numerous Copy and DeleteInstance requests.
- Repack moves instance elements (*file segments*), not necessarily whole instances. If an instance is in one piece on the repacked tape, it will be moved in it's entirety. However, if the instance is spanned, Repack only moves the portion of the instance that resides on the repacked tape; not the other portions residing on other tapes. DIVAmigrate moves the whole instance because it uses the standard Copy function. This makes DIVAmigrate a useful tool to despan instances.
- Repack always writes the source tape content to a single destination tape. DIVAmigrate uses the standard Copy function, and the only guarantee is that the data will end up on the requested destination group. However, not necessarily on a single tape.

Despanning Instances

A spanned instance is an object instance that is written across multiple tapes (*usually two*). Spanned instances require additional tape mount operations during reading, which can cause issues in some contexts. For example, direct restore from tape may experience a timeout on the Server during tape remounting.

You use the following command to despan a series of instances:

```
client -instlist file.txt -media Main -delete
```

Telestream Support can provide you with a utility to generate an instance list file containing the list of instances spanned in the system.

Due to the high capacity capabilities of modern tape units, they tend to suffer from random reduced capacity from intermittent bad block writes. Therefore, the EOT (*End of Tape*) marker may be prematurely encountered. This defeats DIVA Core's remaining tape space calculations and causes accidental spanning.

Using Alternate Source Server Instances

DIVAmigrate always defaults to using the specified Source Server instances to perform the destination copies, or the buffer copy if a *Disk Buffer* is used.

- If an *Object* or *Object List* is used as the Source Server specifier (*-tape or -tapelist option respectively*), DIVAmigrate uses the instances located on these tapes.
- If an *Instance List* is used (*-instlist option*), DIVAmigrate uses the instances listed in the file.
- If an *Object List* is used (*-tapelist option*), DIVAmigrate identifies an optimal instance list and uses that instance.

There are situations where allowing DIVA Core to choose the Source Server instance is desirable. For example, if you are migrating data off of a suspicious tape and some objects possibly still have instances on a storage disk. By default, DIVAmigrate will read the Source Server instances from tape. If you want the Manager (*rather than DIVAmigrate*) to choose the Source Server instance (*which will select an existing disk instance*), you use the -autosrc option as follows:

```
client -tape 000001 -media Main -delete -autosrc
```

The -delete option always deletes the Source Server instances obtained from the Source Server specifier in the command line (*-tape, -tapelist, or -instlist*), regardless of which particular object instance was actually selected as a copy Source Server by the Manager. For example, if you specify -tape 000001 on the command line with the -autosrc option, and one of the objects has a disk instance, this object is copied from disk not from tape. However, the instance deleted after the run is still the one located on tape 000001.

Excluding Objects from Migration

If you observe repeated read errors on some Source Server instances, and you do not want to delete them at this point, you can exclude them from the migration process using the -excl option as follows:

```
client -tape 000001 -media Main -delete -excl file.txt
```

This command ignores all instances belonging to the objects listed in the file named file.txt. This plain text file lists objects to be excluded in the form object|category, one entry per line. For example:

```
Clip001|HD
Clip002|SD
```

9

Monitoring and Error Handling

This chapter describes DIVA Core monitoring and DIVAmigrate error handling and includes the following sections:

- Request Warnings
- Backup Service Warnings and Notifications
- Export/Import Error Handling and Failure Scenarios
- DIVAmigrate Error Handling and Failure Scenarios
  - Migration Error Handling
  - Migration Failure Scenarios

During normal operations, you must periodically monitor the ***Errors*** column in the Manager Current Requests view for warnings and/or errors.

An orange exclamation mark indicates that the request had recoverable errors.

A red exclamation mark indicates that the request had an irrecoverable error and was terminated.

When a system notification is generated (*a warning or error*), a large red X icon is displayed in the bottom right of the Control GUI. If there are no errors the X icon will be gray. Double-clicking the icon automatically changes the current screen to the **Events** view.

Contact Telestream Support for additional assistance when required.

Request Warnings

A warning status indicated on a request signifies that an unexpected error occurred during the requests execution, but the request was still completed.

The following are three example Scenarios:

An I/O error occurred when reading an object from tape. However, there was a second instance of the object on another tape. DIVA Core attempted to use the second instance and this time the object transferred successfully. You must attempt to investigate the tape from the first restore attempt. If multiple events of this type occur across multiple tapes, you must establish whether they all relate to a specific tape drive. If the errors are severe, DIVA Core will automatically mark the drive **Out of Order**.

An object is being transferred to a disk array. Because multiple disks can be assigned to an array, an unexpected I/O error may have occurred with one of the disks in the array. DIVA Core automatically selects another disk from the array to transfer the object to, and this attempt is successful. The disk where the I/O error occurred is marked **Out of Order** by DIVA Core and not used again. The offline disk must be examined for the cause of the error.

An object is being archived to tape and a write error occurs with the selected tape. DIVA Core attempts to use another tape and drive to fulfill the request. The tape from the first write attempt is marked **Read-Only**, and not used for additional archive requests.

Backup Service Warnings and Notifications

In the DIVA Core 8.1 release the Backup Service error and warning dialog boxes are no longer displayed in the Control GUI.

Export/Import Error Handling and Failure Scenarios

This section describes basic export and import troubleshooting procedures.

Export Failed Error Message

```
Robot Manager Error : Error while ejecting tapes: StatusCode[70:INTERNAL_ERROR]
Request step is STEP_WAITING_FOR_OPERATOR()
```

Resolution:

Check that the CAP where tapes are being ejected to has not reached its capacity. Even if the CAP is empty, if more tapes than the capacity of the CAP are exported a successful export operation cannot be completed. This is specifically an issue with sets of spanned tapes and the number of tapes in that spanned set is greater than the number of tapes supported by the CAP. In this case, eject the tapes first then perform the export.

Invalid Parameter Error During Export

```
Invalid parameter : Tape Y00105 must be included into export list
```

Resolution:

When selecting the tapes for export, you can possibly see more tapes available in the tape window than initially selected. If a tape has objects that are spanned onto another tapes, these tapes are also included. In this case, select all of the spanned tapes from this list in order for the export to succeed.

Tape Already Exists Error During Import

```
The following errors were found in tapeset-J00026.xml\Tape J00026 already exists in DIVA. Consider performing a tape
Insert operation...
```

Resolution:

A tape with the same barcode as the one being imported already exists in the DIVA Core system. It is likely that the tape metadata for the tape you want to import already exists in the DIVA Core database and you just need to perform an **Insert Tape** operation to use the tape. Verify the tape contains the correct objects by using the Control GUI.

Unsupported Type Error During Import

```
The following errors were found in tapeset-[Y00109].xml\Tape Y00109 has unsupported type 19.
```

Resolution:

The *type* in the message refers to the mediaTypeId. The mediaTypeId is an ID that represents the type of tape media being exported. DIVA Core exports a mediaTypeId field that corresponds to the ***Id*** column in the ***Tape Properties*** table under the **Tapes** tab in DIVA Command. You may need to execute a Synchronize DB call to update the mediaTypeId and (*or*) update your hardware to be compatible with a newly imported tape. Ensure that the block size and total size of the mediaType in the source DIVA Core system matches the mediaType definition in the destination.

Import Process Terminated without Importing

There are several reasons why the import process may terminate without completing successfully including the following:

- When using complex objects, the FFM files must be in the same folder as the XML files for importing. If the FFM file is not found, the import process will terminate and an error will be written to the log file.
- If the *Object Name* and *Object Category* already exist, and the -skipIfNameExists or -addAsInstanceIfNameExists options are not passed, the utility will terminate without importing.
- If the Manager detects a database error the import process will be terminated and any operations performed during the failed import will be rolled back and not saved in the system.

DIVAmigrate Error Handling and Failure Scenarios

This section describes error handling and possible failures that may occur during migration and includes the following information:

- Migration Error Handling
- Migration Failure Scenarios

Migration Error Handling

DIVAmigrate performs error handling based on following process:

1. If the configured number of request failures occur sequentially, the migration job attempts to switch to another ***Migration Destination***. The job is paused if there are no other destinations.
2. After failing for the configured number of request attempts for a destination, the migration job switches to another destination. DIVAmigrate then attempts to process the job using the new destination until the configured number of requests fails sequentially. If this attempt also fails, DIVAmigrate attempts to switch to the next destination (*if any exist*). After all destinations have been exhausted, the job will pause (*if no more destinations exist*).

   The only exception to this rule is failed requests due to a too many running requests error; these requests are always retried.

In general DIVAmigrate does not retry migration for objects. DIVAmigrate stores the error codes for each object, and moves on to next object requiring migration. You can rerun the migration job later, and it only affects new objects on the Source Server, and those that failed during the previous run; objects successfully migrated the first time are not retried when you rerun the failed job.

Migration Failure Scenarios

This section describes specific failure scenarios, actions by DIVAmigrate to resolve the failure, and suggested actions for you.

Table 9–1 DIVAmigrate Failure Scenarios and Resolutions

Failure	DIVAmigrate Action	Suggested User Action
DIVAmigrate lost the connection to the Manager.	Tries to reconnect until connection is restored.	If the Manager is offline or not running, then restart the Manager. If DIVAmigrate cannot connect to the Manager, you can identify it because new migration jobs stay too long in the **Submitted** state, and do not change to the **Pending** state. You can also check the DIVAmigrate Logs. At this point you should investigate common problems with the connection.
DIVAmigrate cannot connect to the Manager Database.	Retry the connection three times every 10 seconds; if not successful then quit.	You must investigate and resolve common database connectivity issues, and then restart the DIVAmigrate service.
The environment where DIVAmigrate is running fails.	After restart, DIVAmigrate checks the status of each job in the database. If they are not expired, it continues their processing. Otherwise, DIVAmigrate recreates their migration plan and resumes job processing as soon as free resources are available.	You must resolve the operating system environment issue, and then restart DIVAmigrate.
There are no online instances for an object on the ***Migration Source***.	DIVAmigrate checks for online instances only during the *Initialization* phase. If an object does not have any online instances, DIVAmigrate skips it and assign the correct error code to the object in the database. If the object was online, but later became offline during the *Initialization* phase, then some requests sent to the Manager will fail. The error code is stored in the database, and the request is not retried. If the requests continuously fail up to the maximum number of failures, the job pauses. If the job has multiple Source or Destination Servers, it switches to the next server. If alternative instances migrations fail, the original instances are used instead.	You can browse the results of migration for each completed migration job. The results contain a list of all objects eligible for migration, and resulting error codes (*if any errors occurred*). You should make objects accessible online again, and then rerun the migration job.

Table 9–1 (Cont.) DIVAmigrate Failure Scenarios and Resolutions

Failure	DIVAmigrate Action	Suggested User Action
You perform a migration from a tape group that has offline tapes.	If the ***Alternative Sources*** option is turned on, DIVAmigrate attempts to locate disk instances for all offline instances, and then migrates them. If ***Recovery Mode*** is turned on, DIVAmigrate attempts to locate disk or tape instances for offline Source Server instances, and then migrates them. If no online alternative instances are found for some offline instances, they are skipped from migration. An *Object Offline* error code is assigned to the objects. If the tape transitions to offline during migration, the job pauses after the maximum sequential failure requests. If the job has multiple Sources or Destination servers, it switches to the next one, and then pauses after processing them.	You can browse the results of migration for each completed migration job. The results contain a list of all objects eligible for migration, and error codes (*if any errors occurred*). You must make objects accessible online again, and then rerun the migration job.
You submitted the migration job with invalid parameters.	DIVAmigrate assigns the **Aborted** status to a job.	You must confirm the job parameters, and then create a new migration job with correct parameters.
A user deletes unmigrated objects from the Source Server during a migration procedure.	DIVAmigrate attempts to migrate the objects, and stores the appropriate error codes in the database.	You should use the Control GUI to **Cancel** or **Delete** the migration job.
You realize that you submitted a migration job with incorrect parameters.	Not applicable	You should use the Control GUI to **Cancel** or **Delete** the migration job.
There is not enough free space on the Destination.	DIVAmigrate fails the configured maximum number of failed requests sequentially, and then places the job in the **Paused** state.	You must resolve the issue on the destination, and then resume the job.
All ***Migration Destination*** media goes offline.	DIVAmigrate fails the configured maximum number of failed requests sequentially, and then places the job in the **Paused** state.	You must resolve the issue on the destination, and then resume the job.
The *Source Tape List*, *Instance List*, or *Exclude List* has invalid values, but the syntax is correct. For example, a tape barcode is invalid.	DIVAmigrate ignores invalid values. If all of Source Server is invalid, the job must terminate.	Confirm the values entered into these lists, make any necessary corrections, and then rerun the job.

10

Operational Boundaries

Number of Manager Connections

The number of connections to the Manager is limited by the Manager and set in the Manager configuration file. The default configuration limit is two hundred. This boundary includes connections to GUIs, Datahubs and all API clients. When the configured limit is reached, the API will not create additional connections.

See the manager.conf file for more information.

Number of Simultaneous Manager Requests

The maximum number of simultaneous requests processed by the Manager is configurable in the manager.conf file as the value of the DIVAMANAGER_MAX_SIMULTANEOUS_REQUESTS parameter. The default value has been raised from two hundred to five hundred in DIVA Core 8.0 (*and later*). The maximum number has been verified up to two thousand. Additional simultaneous requests beyond the value set in this parameter are rejected by the Manager.

Number of API Tasks

The number of API tasks that will be accepted to the API Processing Queue is configurable in the manager.conf file as the value of the DIVAMANAGER_API_TASK_QUEUE_SIZE parameter. The default value is two thousand and DIVA Core 8.1 has been verified at this value. If the queue is full subsequent commands are rejected.

Recommended API Connection Use

Telestream recommends that a new connection between the Manager and an API client *not* be created for each request or command sent to the Manager. Whenever possible, let the connection remain open for the lifetime of the session or application.

Special Authorized Characters

Many requests require alpha numeric text parameters. Special characters can be used in these fields as defined in the following table. The request is rejected if an invalid special character is used. In a Windows environment, file and folder names cannot consist of one or more spaces, and cannot contain a double-quotation mark.

Table 10–1 Special Authorized Characters in DIVA Core

Field (*across*) **Character (*down*)**	**Name**	**Category**	**Source**	**Media**	**Path**	**File**	**Comments**	**Options**
~	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
'	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
!	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
@	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
#	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
$	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
%	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
^	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
&	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
*	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
(	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
_	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
+	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
=	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
|	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
\	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
}	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
{	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
[	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
:	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
;	Yes	Yes	Yes	Yes	Yes[1]	Yes	Yes	Yes
“	Yes	Yes	Yes	Yes	No	Yes	Yes	No
'	Yes	Yes	No	No	Yes[1]	Yes	Yes	Yes
<	Yes	Yes	Yes	Yes	No	Yes	Yes	No

Table 10–1 (Cont.) Special Authorized Characters in DIVA Core

Field (*across*) Character (*down*)	Name	Category	Source	Media	Path	File	Comments	Options
,	Yes	Yes	Yes	Yes	Yes[1]	Yes	Yes	Yes
>	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
.	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
?	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
/	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Space	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes

[1] Depends on file system restrictions.

Maximum Number of Allowed Characters

The maximum number of characters that can be used for request parameters is displayed in the following list. If these limits are exceeded, the request will be rejected.

Name
Maximum of 192 characters.

Category
Maximum of 96 characters.

Source
Maximum of 96 characters.

Media
Maximum of 96 characters.

Path and File Name
Maximum of 1536 characters.

Comments
Maximum of 4000 characters.

Options
Maximum of 768 characters.

File Path Limitations

DIVA Core 8.1 supports absolute path names on both Windows and Linux up to a maximum of 4000 characters. Relative path names are limited to 256 characters on Windows systems (*only*).

A DIVA Core Windows local path is structured in the following order and terminated with a NUL character:

```
Drive_Letter:\Component_Name\Component_Name\File_Name.Extension
```

A DIVA Core Linux local path is structured in the following order:

```
/Component_Name/Component_Name/File_Name.Extension
```

The following are several example paths used in DIVA Core - Windows first and then Linux. The <NUL> character used in the example represents the invisible terminating null character for the current system code page and need not be typed in. The < and > characters are used in the examples for clarity only and must not be part of a valid path string.

Generic Path:

```
D:\Some_256-Character_Path_String<NUL>
```

```
/Some_4000-Character_Path_String
```

Datahub Executable:

```
D:\diva\diva\Program\Actor\bin\DIVA Core.exe
```

```
/home/diva/diva/std_linux
```

Manager Configuration:

```
D:\diva\74\Program\conf\manager\manager.conf
```

```
/home/diva/DIVA/Program/conf/manager/manager.conf
```

Amazon S3 Bucket Limitations

A given AWS account cannot create more than 100 buckets. This limit can be optionally increased to 1000. Therefore, the number of objects that DIVA can archive per bucket has been increased to 100,000 AXF objects per bucket to enable more DIVA instances per cloud account.

Note: DIVA cannot be configured to write an unlimited number of objects in a given bucket using -bucket_name in the ***Storage Option*** setting.

11

Frequently Asked Questions

In general, refer to the documentation for the specific component for Frequently Asked Questions about that particular component. This chapter includes some basic examples from those documents and answers the following questions:

- Questions and Answers
- Export/Import Questions and Answers

Questions and Answers

This section includes frequently asked questions about DIVA Core Operations.

How do I access the system in Engineering mode?

Contact Telestream Support to access the system in Engineering Mode. Engineering Mode is accessible for Telestream Support personnel only to avoid accidental misconfiguration of the system, which could possibly result in degradation of operations.

What is the Administrator log in and password?

Contact Telestream Support for this information.

How often are metrics updated?

Data Metrics are calculated and updated every hour through an automated database job that runs in the background.

What should be done if SPM is not working as desired?

First, confirm the SPM configuration. *Refer to the SPM (Storage Plan Manager) Guide for details*.

Check the SPM Actions panel in the GUI to confirm that the actions are changing states (***In Process, Completed, Failed****, and so on*).

Check the SPM log file for new entries.

Collect the system logs and database dumps using the Customer Information Collection Tool and submit them to Telestream Support. *Contact Telestream Support for assistance using the Customer Information Collection Tool*.

Will an S3 interface be implemented directly to an S3 Object and S3 Glacier, or will there be something between DIVA and storage?

S3 integration is similar to the OCI integration as it is direct to the object storage. There is nothing in between the Datahub and S3 buckets. Telestream will provide support for all storage classes, including Standard, Standard-IA, One Zone IA, Intelligent Tiering, Glacier and Deep Archive.

For Glacier, there are 3 tiers of retrievals: Expedited (*1-5 min*), Standard (*3-5 hours*) or Bulk (*days*). It appears that everything defaults to Standard, but you can make a restore expedited by changing the Tier parameter of the job request. Is this something that will be supported by DIVA, or will everything be standard?

DIVA Core 8.1 supports the Expedited tier and the default tier (*that is, Standard*).

How will PFR work in an AWS environment assuming the customer is using almost exclusively Glacier?

Like any request from Glacier or Deep Archive, DIVA Core will stage the content to Standard prior to transferring from AWS.

We to want to replicate between two regions but we want AWS to do it and not DIVA. For example, to archive content to N. Virginia and then use AWS to move it to a second region, for example, Ohio. Assuming we have buckets configured correctly in both locations, after it is moved, will DIVA be able to discover the content that is now in Ohio?

On archive, DIVA creates a data and metadata bucket and stores AXF files corresponding to the customer's content in those buckets. The newly created disk instance points to the newly created buckets. If the user were to replicate the AXF files from one region to another through AWS, the replicated content would be placed in a uniquely named bucket in the other region. DIVA would not know the name of the new bucket, and therefore would not be able to reference it. However, when we add support for S3 in our new Disk Discovery service, it should be possible to scan all buckets in an account and add the content as a new instance.

For OCI, we were using Standard 2.4 VM shapes (*4 OCPU, 60GB RAM, 1 TB block storage*) for Datahubs and Managers. Do you think something equivalent will work on AWS? It appears that maybe an 'i3en.xlarge' would be close with 4 vCPU, 32 GB RAM, 1x2.5TB SSD?

Telestream believes this configuration would work fine.

Export/Import Questions and Answers

This section includes frequently asked questions about Export/Import Operations.

What is the export XML and FFM file compatibility?

The exported XML and FFM files, when generated, can be imported into the release of DIVA Core that they were exported from, and later releases of DIVA Core. DIVA Core enables more than one set of tapes (*spanned or not*) to be exported to and imported from a single file.

Exported metadata from the DIVA Core 7.7 export function cannot be imported into DIVA Core releases earlier than release 7.0. However, exported metadata created from releases of DIVA Core before 7.7 can be imported into the DIVA Core 7.7 system.

What is the Media Type ID?

The Media Type ID is a proprietary DIVA Core identifier that represents the type of tape media being exported. DIVA Core exports a mediaTypeId field, which corresponds to the ***Id*** column in the **Tape Properties** table under the **Tapes** tab in the DIVA Command. You may need to execute a Synchronize DB call to update the mediaTypeId, and (*or*) update your hardware to be compatible with a newly imported tape. You should ensure that the block size and total size of the mediaType in the source DIVA Core system matches the mediaType definition in the destination. This becomes especially important if the tape is ever repacked.

What are the unsupported DIVA Core attributes?

The markedAsDeleted is an internal attribute and is not exported or imported through the Export/Import Utility. In addition, the state of checksum verification (*verified, partially verified, and so on*) is not exported. Linked objects and link information is not exported. Information regarding the request that created each object is not exported - newly imported objects are not associated with a request.

A
Appendix

This appendix includes the following:

- Repack and Verify Tape Request Limitations with Checksum Workflows
- Example Non-Spanning Export XML File
- Example Spanning Export XML File
- Sample BKS Configuration File
- Sample DBAgent Configuration File

Repack and Verify Tape Request Limitations with Checksum Workflows

The repack request will fail if the tape contains any corrupted objects or the object fails checksum verification. You must manually resolve the conflict before performing the repack.

Repack of WORM media is not automatic. Manual repack is available for WORM media, but the space is not recoverable after repack is complete.

A checksum is not generated for any spanned objects during Repack or Verify Tape requests. The Datahub will identify any spanned files and the Manager will not attempt to verify them. A warning event will be displayed stating that a checksum was not generated or verified for the spanned content. In this case the repack operation will not be terminated, but the object instance will be marked as **Not Verified**.

Additional checksum verification is done at the Oracle Storage Cloud level. *See the Storage Cloud documentation for information*.

Example Non-Spanning Export XML File

```
<tapeset class="com.storagetek.diva.messaging.types.ExportedTapeSetMetadata" exportDate="27 Oct 2010 20:55:30
GMT" divaName="MGR_650" divaVersion="DIVA_6_5_1_0_0">
 <tapes array-size="1">
 <tape barcode="Y00103" mediaTypeId="13" remainingSizeKB="30803" fillingRatio="3" fragmentation="0"
blockSize="65535" lastWrittenBlock="19" lastArchiveDate="27 Oct 2010 20:55:01 GMT" firstInsertDate="21 Apr 2010
19:02:49 GMT" firstMountDate="27 Oct 2010 20:54:05 GMT" isHeadTape="true" originalGroup="MOV">
  <elements array-size="4">
   <element objectName="TEST" category="SMALL" compNum="1" elemNum="1" beginPos="2" endPos="5"
elemSizeKB="2" stopPos="2371" />
   <element objectName="TEST2" category="SMALL" compNum="1" elemNum="1" beginPos="7" endPos="10"
elemSizeKB="1" stopPos="41" />
   <element objectName="TEST3" category="SMALL" compNum="1" elemNum="1" beginPos="12" endPos="15"
elemSizeKB="1" stopPos="73" />
   <element objectName="TEST3" category="SMALL" compNum="2" elemNum="1" beginPos="16" endPos="17"
```

```
elemSizeKB="1" stopPos="72" />
   </elements>
   </tape>
  </tapes>
 <objects array-size="3">
  <object objectName="TEST" category="SMALL" comments=" " sourcename="origin_ftp" rootOnSource=" "
dateArchive="27 Oct 2010 20:54:05 GMT" numComponents="1" numElements="1">
   <components array-size="1">
   <component name="a1.txt" compNum="1" sizeKB="2" sizeBytes="2372">
   <checksums array-size="1">
    <checksum csValue="40f818c93e17c94fd476951f9f5db788" csSource="AC" csType="MD5" />
   </checksums>
   </component>
  </components>
  </object>
 <object objectName="TEST2" category="SMALL" comments=" " sourcename="origin_ftp" rootOnSource=" "
dateArchive="27 Oct 2010 20:54:20 GMT" numComponents="1" numElements="1">
  <components array-size="1">
  <component name="a2.txt" compNum="1" sizeKB="1" sizeBytes="42">
   <checksums array-size="1">
   <checksum csValue="0be6e7d72fdb52266b9c99540b3755ce" csSource="AC" csType="MD5" />
   </checksums>
  </component>
 </components>
 </object>
 <object objectName="TEST3" category="SMALL" comments=" " sourcename="origin_ftp" rootOnSource=" "
dateArchive="27 Oct 2010 20:55:01 GMT" numComponents="2" numElements="1">
  <components array-size="2">
  <component name="a3.txt" compNum="1" sizeKB="1" sizeBytes="74">
   <checksums array-size="1">
   <checksum csValue="b0354657e98cf78074a6409dce2697c8" csSource="AC" csType="MD5" />
   </checksums>
  </component>
  <component name="a4.txt" compNum="2" sizeKB="1" sizeBytes="73">
  <checksums array-size="1">
   <checksum csValue="2bfa170db4ada38a27085cb4b339f05e"csSource="AC" csType="MD5" />
   </checksums>
  </component>
  </components>
  </object>
 </objects>
</tapeset>
```

Example Spanning Export XML File

```
<tapeset class="com.storagetek.diva.messaging.types.ExportedTapeSetMetadata" exportDate="27 Oct 2010 20:44:57
GMT" divaName="MGR_650" divaVersion="DIVA_6_5_1_0_0">
 <tapes array-size="2">
  <tape barcode="Y00105" mediaTypeId="13" remainingSizeKB="500" fillingRatio="98" fragmentation="0"
blockSize="65535" lastWrittenBlock="500" lastArchiveDate="27 Oct 2010 20:38:59 GMT" firstInsertDate="21 Apr 2010
19:02:49 GMT" firstMountDate="27 Oct 2010 20:38:55 GMT" isHeadTape="true" spannedTo="Y00104"
originalGroup="MOV">
   <elements array-size="1">
    <element objectName="BIG2" category="SPAN" compNum="1" elemNum="1" beginPos="2" endPos="500"
elemSizeKB="31679" stopPos="32440080" />
   </elements>
  </tape>
  <tape barcode="Y00104" mediaTypeId="13" remainingSizeKB="14360" fillingRatio="55" fragmentation="0"
blockSize="65535" lastWrittenBlock="280" lastArchiveDate="27 Oct 2011 20:38:59 GMT" firstInsertDate="21 Apr 2010
```

```
19:02:49 GMT" firstMountDate="27 Oct 2010 20:38:59 GMT" isHeadTape="false" originalGroup="MOV">
   <elements array-size="1">
    <element objectName="BIG2" category="SPAN" compNum="1" elemNum="2" beginPos="2" endPos="278"
elemSizeKB="17443" stopPos="50302194" />
   </elements>
  </tape>
 </tapes>
 <objects array-size="1">
  <object objectName="BIG2" category="SPAN" comments=" " sourcename="origin_ftp" rootOnSource=" "
dateArchive="27 Oct 2010 20:38:59 GMT" numComponents="1" numElements="1">
   <components array-size="1">
    <component name="Dbig.txt" compNum="1" sizeKB="49122" sizeBytes="32440081">
     <checksums array-size="1">
      <checksum csValue="f53d6dbdaa266a5e7327683f971fcd7d" csSource="AC" csType="MD5" />
     </checksums>
    </component>
   </components>
  </object>
 </objects>
</tapeset>
```

Sample BKS Configuration File

The following is a sample Backup Service configuration file with field descriptions and is located in $DIVA_HOME\Program\conf\db_agent\appsettings.json:

```
{
  // Configures the logging level
  "Logging": {
    "LogLevel": {
      "Default": "Debug", // Use this field for general logging levels.
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information",
      "System.Net.Http.HttpClient": "None"
    }
  },
  "Version": "0.0.0.0",
  "AllowedHosts": "*",
  // Configures the databases managed by the backup service.
  // Note: This is configurable via the API
  "DatabaseSettings": {
    "Databases": [
      {
        "ProfileName": "MetadataDatabase", // Profile for a specific DB instance
        "DatabaseName": "Core", // Name of the database itself (used for failover)
        "DatabaseType": "MongoDB", // Type of DB (MongoDB, Oracle, Postgres, ElasticSearch)
        "DatabaseVersion": "5.0", // DB Version
        "ConnectionString": "mongodb://127.0.0.1:27017/?replicaSet=rs0", // Connection String
        "RootDirectory": "", // Root directory to controlling applications (DB dependant)
        "User": "MongoAdmin", // Admin DB user
        "Password": "some password"
      },
      {
        "ProfileName": "OracleDatabase",
        "DatabaseName": "lib5",
        "DatabaseType": "Oracle",
        "DatabaseVersion": "12.1.0",
        "ConnectionString": "",
        "RootDirectory": "",
        "User": "diva",
      "Password": "some password"
    }
  ]
},
// Configures the backup locations to be replicated against and Agent api's associated with them.
"LocationSettings": {
  "Locations": [
    {
      "Name": "Primary", // Name of the location
      "Primary": true, // Whether that location is the primary (all copies come from the primary location)
      "Enabled": true, // Whether the location is enabled
      "Location": "H:\\divaback", // The path to the location
      "AgentUrl":  "https://localhost:1878/", // DBAgent RESTapi Url
      "Type": "Local", // Location type (either Local or UNC)
      "ManagedDatabases": [ // List of managed databases
        "OracleDatabase",
        "MetadataDatabase"
      ],
      "SourceName": "" //This is the name of the source in DIVA Core for backup archives.
    },
    {
      "Name": "Secondary",
      "Primary": false, //There may only be one primary location
      "Enabled": true,
      "Location": "\\\\100.10.10.10\\H$\\divaback",
      "AgentUrl": "https://100.10.10.10:1878/", //Agent URL must resolve to the remote location
      "Type": "UNC",
      "ManagedDatabases": [],
      "SourceName": "",
      "User": "Administrator", //UNC paths must have a username and password
      "Password": "some password"
    }
  ]
},
// Configured the API timeouts and session expiration.
"ServiceSettings": {
  "RequestExpiration": 3600,
  "RequestTimeout": 600
},
// Configures the API endpoints
"HttpServer": {
  "Endpoints": {
    "Http": {
      "Host": "localhost",
      "Port": 1876,
      "Scheme": "http"
    },
    "Https": {
      "Host": "localhost",
      "Port": 1877,
      "Scheme": "https",
      "FilePath": "../../security/certificates/BackupService.p12" // Cert path for the https endpoint.
    }
  }
},
// Controls the endpoint for reporting to the DIVA API Gateway
"DiscoverySettings": {
  "Url": "https://127.0.0.1:8761/eureka/apps/",
  "AppName": "backup-service"
},
  // Configures the endpoint to report statuses and make archive requests.
  // Note: This is configurable via the API
  "DIVACoreAPISettings": {
    "Url": " https://127.0.0.1:8765/",
    "User": "Some User",
    "Password": "Some Password",
    "TimeoutInMs": 20000 // Time in milliseconds to wait for a request to complete.
  },
  // General Service Configuration
  // Note: This is configurable via the API
  "DatabaseBackup": {
    "Enabled": true, // Enables or disables all backups
    "BackupTime": "00:00:00", // Time of day to backup
    "IncrementalPeriod": 15, // Number of minutes to wait to do an incremental backup.
    "FullBackupFileRetention": 7, // Number of days to retain full backups on disk.
    "FullBackupArchiveRetention": 30, // Number of days to retain full backups in archive.
    "ArchiveMediaGroup": "", // The tape group in DIVA Core to archive the backups to.
    "BackupExecutionTimeout": 120, // Number of minutes before timing out a backup request (2 hr default)
    "RestoreExecutionTimeout": 120 // Number of minutes before timing out a restore request (2 hr default)
    "StatusPollingPeriod": 3, // Number of seconds between polling the status of backups
    "StatusReportingInterval": 1440 // Number of minutes to suppress duplicated alerts
  }
}
```

Sample DBAgent Configuration File

The following is a sample DBAgent configuration file with field descriptions:

```
{
  // Configures the logging level
  "Logging": {
    "LogLevel": {
      "Default": "Debug", // Use this field for general logging levels.
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information",
      "System.Net.Http.HttpClient": "None"
    }
  },
  // Configures the API endpoints
  "HttpServer": {
    "Endpoints": {
      "Https": {
        "Host": "localhost",
        "Port": 1878,
        "Scheme": "https",
        "FilePath": "../../security/certificates/DBAgent.p12" // Cert path for the https endpoint.
      }
    }
  },
  "ServiceConfiguration": {
    "BasePath": "H:\\divaback", //Base directory for the backup replication point
    "MountPointMonitors": [
      {
        "Path": "H:\\divaback", //Path to monitor
        "IsPercentBased": true, // Either warn/error on percentage or free byte threshold
        "ErrorThreshold": 0,
        "WarnThreshold": 0,
        "ErrorPercentage": 95,
        "WarnPercentage": 85
      },
      {
        "Path": "C:",
        "IsPercentBased": true,
        "ErrorThreshold": 0,
        "WarnThreshold": 0,
        "ErrorPercentage": 95,
        "WarnPercentage": 85
      },
      {
        "Path": "E:",
        "IsPercentBased": true,
        "ErrorThreshold": 0,
        "WarnThreshold": 0,
        "ErrorPercentage": 95,
        "WarnPercentage": 85
      },
      {
        "Path": "F:",
        "IsPercentBased": true,
        "ErrorThreshold": 0,
        "WarnThreshold": 0,
        "ErrorPercentage": 95,
        "WarnPercentage": 85
      }
    ]
  }
}
```

Glossary

Action

A predetermined reaction of a metric surpassing a threshold value by one of the variables from its internal state. This term is used in the SPM (*Storage Plan Manager*).

Array

An array designates a collection of disks designated by their name as they are declared in the DIVA Core configuration. A disk name is associated with a mounting point. Archive requests can be submitted with an array as the destination. DIVA Core is responsible for choosing the disk location to write the data when several disks belong to the same array.

AXF (*Archive eXchange Format*)

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

CAP (*Cartridge Access Port*)

The Cartridge Access Port is used for inserting and removing tape cartridges to and from a Robotic Tape Library without interrupting library operations.

CAP ID

The designation of a slot in the Tape Library.

Category

Categories are an approach to grouping an object with other similar objects having particular shared characteristics. It must not be confused with mediums or arrays, which are storage concepts.

Checksum and Checksum Types

A mathematical value computed from a group of data being transmitted, and transferred with the data. The receiving device compares the checksum with its own computation, and if it differs from the received checksum, it requests the transmitting device to resend the data or generates an error. Each checksum has a specific algorithm, each of which has its own level of verification.

Additional checksum verification is done at the Oracle Storage Cloud level. See the Storage Cloud documentation for information.

Complex Object

An Object is defined as a Complex Object when it contains 1,000 (*configurable*) or more components. Complex object handling may differ from non-complex objects as noted throughout this document.

Component

A file that is part of an object.

Destination

A system on which archived objects are restored.

DFM (*Drop Folder Monitor*)

The DFM (*Drop Folder Monitor*) monitors preconfigured drop folders on the system. When new files are detected, one or more operations are performed on the files depending on the folder configuration. *Refer to the DFM (Drop Folder Monitor) User's Guide for more details*.

DET (*Dynamically Extensible Transfers*)

Dynamically Extensible Transfers are an Avid protocol.

Drop Folder

A folder on a local disk, FTP server, or a CIFS shared folder designated for Single File mode or File Set mode that is monitored by DFM, and from which files will operations performed on them.

Event

One operation (*such as a request*) usually requires multiple events to complete an operation. An event provides all applicable information relating to the single task (*for example, names, IDs, parameters, numbers, and so on*).

Externalization

An object instance is ejected when one of the tapes containing the object's instance elements is ejected. An object is externalized when all of its instances are ejected. An object is considered inserted when at least one instance of the object is inserted.

Legacy Format

Proprietary storage format used in DIVA Core releases 1.0 through 6.5.1.

Media Format

Tapes and disks can be formatted as either AXF or Legacy (*format used before release 7.0*) format. The format is set for tape groups and disk arrays during configuration. Complex objects must be stored on AXF-formatted media.

Medium

Set of storage resources. Currently DIVA Core provides two types of media: groups of tapes and arrays of disks. DIVA_archiveObject() and DIVA_copyToGroup() requests transfer to a Medium (*media*).

Metadata Database

The metadata database is the location where the metadata for components of complex objects are stored in the DIVA Core system.

Metadata File

The file listing the *Object Name* and *Object Category* contained on a tape and its location.

Metric

An instance of one Metric Definition for a specific resource can be either enabled or disabled. Each Metric is associated with a specific resource and can receive a flow of measurements from that attached resource.

A metric has an internal state that consists of several numeric values are updated when given new measurements while providing read access to this logically consistent state. Each metric can be used as a measurement value for the state of another metric. The internal state can be reset at any time.

Metric Definition

Defines how a metric is calculated by specifying which events are examined, which measurements are extracted, how they are aggregated (*collection type*), and which resource the aggregation is based on.

See the DIVAprotect User's Guide for more information.

Non-complex Objects

Objects with less than 1,000 files are considered non-complex objects. The maximum number of files an object can hold is configurable.

Object

Objects are archive entries. An object is identified by a pair (*Name and Category*) and contains components. A component is the DIVA Core representation of a file. The components are stored in DIVA Core as object instances.

Object Instance

Mapping of an object's components onto a set of storage resources belonging to the same storage space. Deleting instances cannot result in deleting the related object. The deletion of a unique instance is not permitted.

Request

A request is an operation running in DIVA Core which progresses though steps (*migration, transfer, and so on*) and ends as **Completed**, **Aborted**, or **Cancelled**.

Resource

Used to denote the necessary elements involved for processing requests (*for example, Datahubs, disk, drive, and tape*). A resource is a uniquely identified element of the DIVA Core system. DIVAprotect references them by events and metrics.

Robot Manager

The mechanical tape system used with DIVA Core to insert and eject tapes to and from the tape library.

Set (*of tapes*)

Every tape in a DIVA Core system belongs to one and only one Set. If the tape is not available to DIVA Core, it belongs to Set #0, otherwise it belongs to a set with a strictly positive ID (*for example, Set #1*). Each group is associated with a Set. When the group needs an additional tape, it takes it from its associated Set.

Source Server

A system that produces data to be archived in the DIVA Core system (*for example, video servers, browsing servers, remote computers, and so on*).

Spanning

Splitting an object's component onto several tapes (*usually two*). This can occur when the component size is larger than the remaining size left on the initial tape.

UUID (*Universally Unique Identifier*)

UUID (*Universally Unique Identifier*) uniquely identifies each object created in DIVA Core across all Telestream customer sites except for objects created through **Copy As** requests. An object created using a **Copy As** request will contain the same UUID as that of the source object.