Oracle® DIVArchive

Export / Import User's Guide

Release 7.3

E64040-02

January 2016

ORACLE®

Oracle DIVArchive Export / Import User's Guide, Release 7.3

E64040-02

Primary Author: Lou Bonaventura

Table of Contents

Tables Index

Figures Index

1 Introduction

1.1 Document Purpose and Scope

The purpose of this document is to describe the Export and Import Tape operations using the Oracle DIVArchive 7.3 Control GUI and the Windows command line interface. Administration and Operations personnel should use this guide to follow all of the necessary steps to provide full performance of the DIVArchive Export and Import functions.

1.2 Document Conventions

The following conventions are used with respect to text:

Normal	Standard Text.
Italic	Used to emphasize a term or variable.
Bold	Used to emphasize critical information.
6.1	Refers to a section or sub-section in the document.
`Courier New`	Used for system screen output and system commands.

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

- **`DIVA_HOME`**: The Root Path on the file system where DIVArchive is installed.

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

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

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

1.3 Definitions, Acronyms, and Special Terms

Table 1: Definitions, Acronyms, and Special Terms

Term	Definition
AXF (*or AXF Media Format*)	The Archive Exchange Format (*AXF*) is based on a file and storage media encapsulation approach which abstracts the underlying file system, operating system, and storage technology making the format truly open and non-proprietary. AXF helps ensure long-term accessibility to valued assets, and keeps up with evolving storage technologies.
CAP ID	The designation of a slot in the Tape Library.
Complex Object	An Object is defined as a Complex Object when it contains more than 1,000 components (*configurable*). Complex Object handling may differ from non-Complex Objects as noted throughout this document.
Legacy Format	DIVArchive proprietary storage format used in DIVArchive version 1.0 through 6.5.1.
Metadata Database	The Metadata Database is the location where the metadata for components of Complex Objects are stored in the DIVArchive System.
Metadata File	The file listing the Object Names and Categories contained on a tape and their location.
Non-Complex Object	By default, DIVArchive Object with 1,000 files or less are considered Non-Complex Object. The Maximum number of files a Non-Complex Object can hold is configurable.
Robot Manager	The mechanical tape system used with DIVArchive to insert and eject tapes to/from the Tape Library.
UUID	Universally Unique Identifier is used to uniquely identify each object created in DIVArchive across all Oracle customer sites, except for objects created via **Copy As Requests**. An object created via a **Copy As Request** will contain the same UUID as that of the source object.
WORM	**Write-Once-Read-Many:** describes a data storage device in which information, once written, cannot be modified. This write protection affords the assurance that the data cannot be tampered with once it is written to the device.
XML	**Extensible Markup Language:** a text-based database used to allow for the easy interchange of documents and data on the World Wide Web or between software components (***.xml***).

1.4 Overview

The **DIVArchive Export/Import** feature allows a user to remove one or more tapes from a DIVArchive System and add them to a second DIVArchive System. The Export feature (*on the first DIVArchive Site*) generates Metadata Files that describe each tape selected for export. These files can then be imported into a second DIVArchive System.

Additionally, the export command ejects the selected tapes from any tape library that they may be in at the time. Through importing the metadata, and then inserting the ejected tapes, the archived objects on the exported tapes can be transferred to the second DIVArchive System.

All **Export** functions and the **Insert Tape** function are executed from the DIVArchive Control GUI, while the **Import** function uses the Windows command line interface. DIVArchive allows 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 were imported. There is an option to import an object as an instance of another object that already exists in the DIVArchive Database (*refer to Section 2.2 for details*). The import utility requires specification of a Target Tape Group for newly imported tape objects. The new objects will belong to this Tape Group and not the Tape Group of the DIVArchive System from which it was exported.

The Export/Import functionality is compatible with Complex Objects and has additional fields to take into account advanced formatting and functionality available in DIVArchive version 7.3.

Note: The exported Metadata from the DIVArchive 7.3 Export function cannot be imported into DIVArchive versions prior to version 7.0. However, exported Metadata created from versions of DIVArchive prior to 7.3 can be imported into the DIVArchive 7.3 System.

2 Operations

2.1 Exporting Tapes

The **Export Tapes** function allows one or more tapes containing DIVArchive objects to be exported for use in another independent DIVArchive System (*e.g. at a remote disaster recovery or partner site*).

The metadata of each tape (*i.e. the object names and categories it contains and their location on the tape itself*) for non-Complex Objects are maintained in the DIVArchive Database. The metadata of each tape is also saved to an XML file when the tape(*s*) are exported, and used to transfer the metadata of each tape into the other DIVArchive System's database during the tape(*s*) import operation.

The metadata for Complex Objects is maintained in both the DIVArchive Database and the Metadata Database. When an Export Request is initiated, the Export Utility creates an additional plain text file and assigns an **`.ffm`** extension to the file.

The Export feature checks to see if any of the tapes that have been selected have 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 in order to export the original list of tapes.

Notes:

The Export Tapes command is NOT used for transferring tapes between two or more libraries controlled by the same Oracle DIVArchive Manager. If this is what is required, use the Eject command, move the tape to the desired library and then perform and Insert Tape command.

The default action in the Export feature is to remove the tape metadata from the DIVArchive 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. Optionally, the metadata for the object may be left in the original DIVArchive Database if desired.

Ejected tapes may be exported as well: Ejecting tapes before exporting them is the recommended method when the number of tapes to be exported exceeds the robotic tape library selected Cartridge Access Port (*CAP*) size.

During an Export/Import operation, whether the media is Write-Once or not and whether the media is a cartridge or not is identified in the exported XML file and also imported with this information. The new attributes of the tape element are `isWriteOnce` and `isCatridge` each with a value of either `true` or `false`.

2.1.1 Export Limitations

Limits on Tape Exports are configured in the **Manager.conf** configuration file. There are several configurable parameters as shown in the table below:

Table 2: Tape Export Limitation Parameters

Parameter	Definition	Limits
DIVAMANAGER_MAX_EXPORT_TAPES	The maximum number of tapes allowed in export request. Reloadable in SERVICE mode.	Default value is 10 and the maximum value is 25. **Example:** **DIVAMANAGER_MAX_EXPORT_TAPES=10**
DIVAMANAGER_MAX_EXPORT_ELEMENTS	The maximum number of elements allowed in an export request. Reloadable in SERVICE mode.	Default value is 100000 and the maximum value is 100000. Example: **DIVAMANAGER_MAX_EXPORT_ELEMENTS=100000**

Notes:

It is HIGHLY recommended to only perform 1 export operation at a time – there is a chance of data loss if more than one export operation is running simultaneously.

Do NOT perform 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.

2.1.2 Export Metadata Parameters

Table 3: New Export Metadata Parameters

Parameter	XML Element/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 -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's element.	If exists in DB
`footerEndPos`	Attribute of the element's element.	If exists in DB
`compOrderNumBegin`	Attribute of the element's element.	If exists in DB
`compOrderNumEnd`	Attribute of the element's element.	If exists in DB
`fileFolderMetadataInfo`	Element	Valid for Complex Objects
`fileFolderMetadataInfo-elem`	Element	Valid for Complex Objects
`checksums` and `checksum`	Element	NOT valid for Complex Objects

2.1.3 Exported Tape Metadata Files

When tapes are exported from the DIVArchive System, each tape's metadata is written to an **.xml** file. An additional **.ffm** file is generated for each Complex Object that is exported. 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
```

- e.g.: `Tapeset-000131.xml`

The Root Path where the XML files are saved is defined by the **DIVAMANAGER_EXPORT_ROOT_DIR** parameter in the DIVArchive 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 Object that was exported. This file **must** exist in the same directory as the **.xml** file when importing, as the Import Utility will look for them both in the same location. If the file is missing, the Import process will abort and an error message will be written to the log file.

2.1.4 Export Tapes Procedure

The **Export Tape** Request is initiated using the **Export Tape Button** on the 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 as shown in the figures below. 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 in order for the export to succeed.

Figure 1: Action Tab Ribbon Bar – Export Tape Button

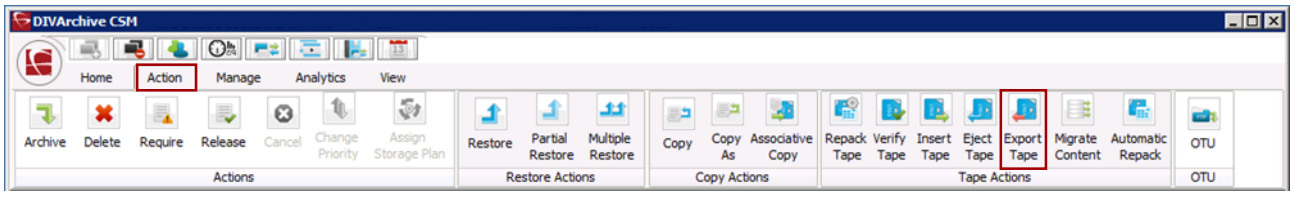
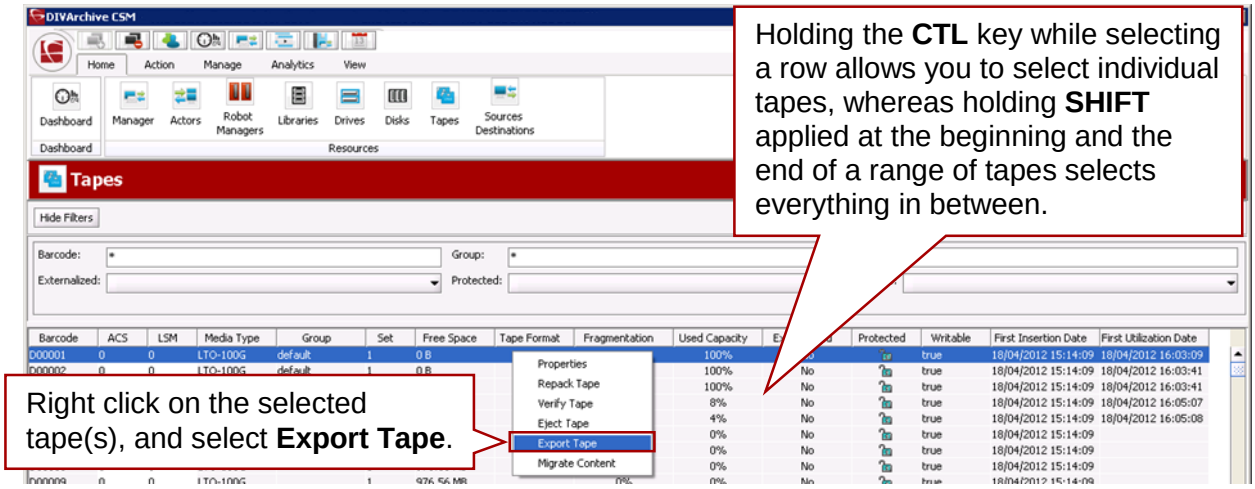


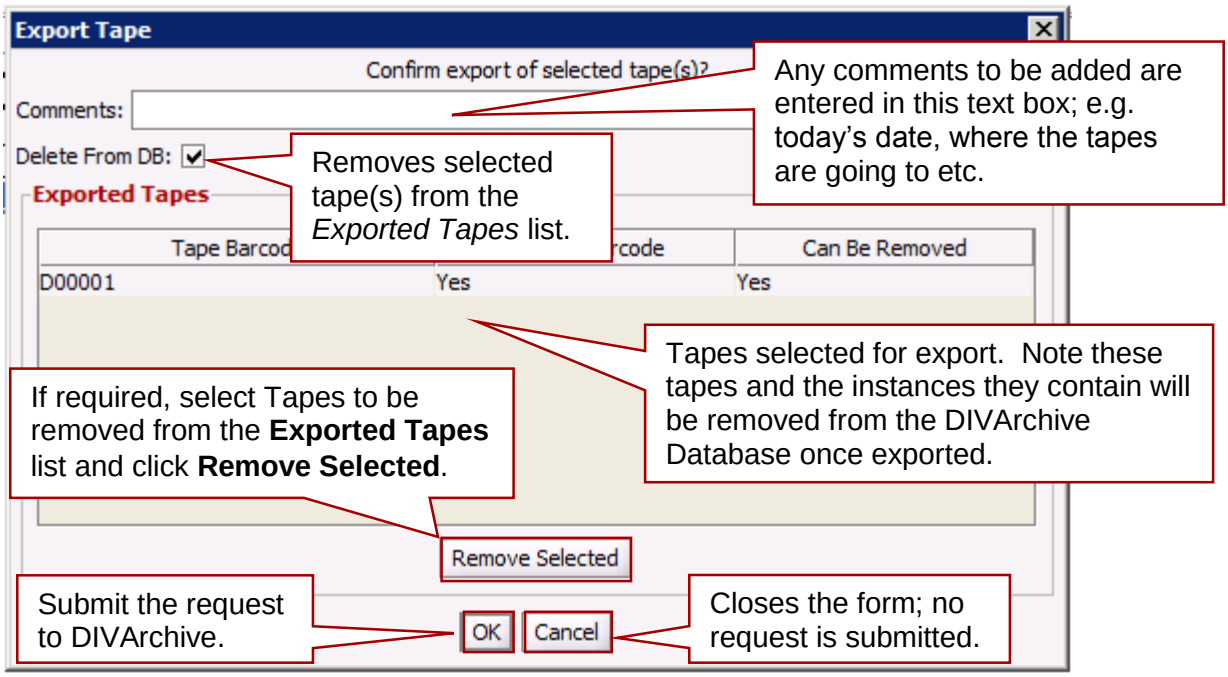
Figure 2: Tapes View Right-Click Menu (Export Tape Menu)



Highlight, and then right-click the tapes desired for export. The pop-up menu seen in the figure above will be displayed.

1. Select **Export Tape** from the menu to begin the export process.
2. The **Export Tape** popup window 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 DIVArchive Database upon completion of the export.
     - If tapes or object instances are needed in the system again after they have been exported they will need to be imported because this option removed them from the system's database.
     - By default this parameter is set to **`true`**.
   - **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 in order to complete the export successfully.
   - **Remove Selected:** Removes the highlighted tapes (*in the Exported Tapes area*) from the export process.

Figure 3: Export Tape Popup Window



3. Once all options have been set and verified, click the **OK** button to begin the tape export.
    a. This is a multi-step process. If a set of tapes was selected that includes another spanned tape, the GUI will display reselection windows allowing for selection of additional tapes in the set.
    b. When the **OK** button is clicked the export process begins, which results in an `.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.
4. The exported files can be found using the Windows Explorer in the `DIVA_HOME\Program\Manager\bin\exported\<date>--<time>` folder.

When the export is complete, it is a good practice to compress all of the resulting files into a `.zip` file. Make sure to include all of the files as they are required for the Import process to complete successfully.

Note: 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 abort and an error will be written to the log file.

2.2 Importing Tapes

Importing tapes so they can 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 DIVArchive Control GUI.

Notes:

Performing multiple simultaneous import operations are allowed; however it is not recommended.

2.2.1 Using the Import Command

To use the **`importtapes`** command, first insure that the XML Metadata file and the **`.ffm`** files that were exported exist on the DIVArchive System into which they are to be imported. The files must exist in uncompressed form (*unzipped*) in the DIVArchive Manager's **`bin`** directory (*by default*). In addition, 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 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
- Added as an instance of an object already existing in the DIVArchive Database.

2.2.1.1 Import as New Object

Normally, when a Tape Object is imported by the utility, it is imported as a new DIVArchive Object. This can only occur when the Object Name/Category for the Tape Object does not exist in the target DIVArchive System. In the event of a naming conflict, the default behavior is to abort the import operation without importing any tapes or objects.

When the new objects are imported into the target DIVArchive System, the import function only looks at the XML and FFM files and does not read directly from the tape structure. Additionally, SPM is automatically notified and if the object matches any of the SPM filters, SPM will initiate the required actions for the object.

2.2.1.2 Skip Object

A tape object can be skipped if the **`-skipIfNameExists`** flag is passed to the Import Utility. If there is another object already in the DIVArchive Database that has the same Object Name/Category as a Tape Object to be imported, and the **`-skipIfNameExists`** flag is set, the Object is *skipped*. This means that the Object Instance on the tape is not recorded in the DIVArchive Database (*it is considered 'deleted' by DIVArchive*), and processing continues with the next Tape Object in the import Metadata.

Note: The Tape Object that is skipped may or may not actually be the same as the Object in the database. Refer to Table 4 for more details.

Care should be taken when skipping objects. The Tape Object that had the naming conflict may in fact contain different content than the one in the DIVArchive 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 writeable*), the tape will be marked for deletion, and new objects will overwrite the existing objects on tape. Finally, if the last Object on a tape is skipped and the new Objects are written to the tape, that tape instance will immediately be overwritten.

2.2.1.3 Using the Import Date as the Archive Date

The DIVArchive **`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 DIVArchive system; only for the object on the imported system.

This feature supports tapes with spanned objects in the same way as regular tapes.

2.2.1.4 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 DIVArchive Database that has the same Object Name/Category as a Tape Object to be imported, and the **`-addAsInstanceIfNameExists`** flag is passed, an import as an instance can be 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, etc. of the imported object are lost and become that of the object already in the DIVArchive Database.

Note: Object instance ids are neither exported nor imported. A new 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 Types in the imported object, or if one of the two objects has checksums that are missing, the Tape Object will not be imported as an instance. This is considered a checksum mismatch and import processing will halt. 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 DIVArchive Database*), the utility will first try to import the object as an instance by comparing checksums. If this attempt fails, the object will be skipped and processing will continue.

Note: SPM is not notified when importing as an instance, and if the object matches any of the SPM Filters, SPM will not initiate the required actions for the object.

2.2.1.5 Error Conditions

If the tape media is not recognized by the Manager an error will be generated specifying what occurred. Refer to Section 3.3 for an example error when the tape media is not recognized.

If the import process fails and Manager detects a database error, the import process will be aborted 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, the operator should check the screen messages and/or log file to determine whether or not all content intended to be imported was processed successfully. Please note that this option is not compatible with automated workflows since it may require operator intervention and decision.

Refer to Section 3.3 for error troubleshooting information.

2.2.1.6 Warnings and Limitations

Complex Objects that are compared this way must have been archived in the same exact order in order to pass the checksum verification.

The Import Utility does not compare UUIDs, Object IDs, Archive Dates, or Site IDs. The Comments, archived Path Root, Archive Date, UUID, Storage Plan, Group, etc. of the imported object are **not** preserved when being added as an instance.

The utility also does not allow 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 should prevent this from happening.

2.2.2 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 **`Tapeset-000131.xml`**, which will also include 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 DIVArchive Database.

The XML File is then copied to the **`DIVA_HOME\Program\Manager\bin`** folder of the DIVArchive System on which it is to be imported.

The metadata for this tape will be imported into the group **MOVIES**:

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

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 then considered externalized, and can then both be entered into the library with the **Insert Tape** command.

Note: Importing of WORM Media is supported by DIVArchive 7.3 and later; however, if an import of a DIVArchive 7.3 (*or later*) export that contains WORM Media into an older DIVArchive version (*pre-7.3*), the WORM will be ignored (*set to `false`*) and logged 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.

2.2.3 Import Tape Procedure

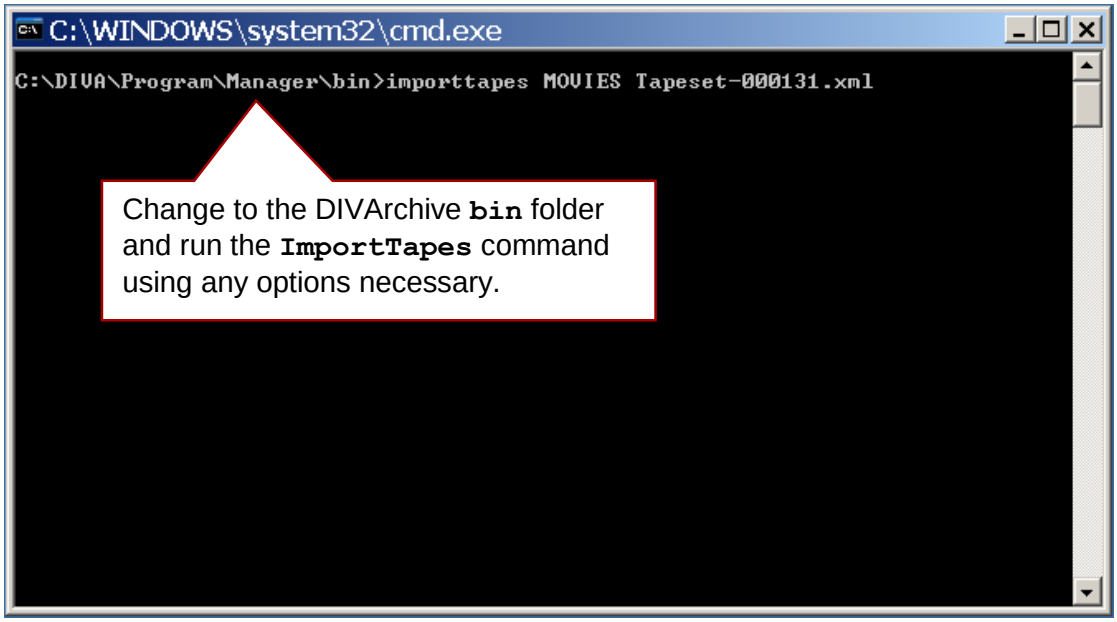
Importing of tapes is accomplished using a combination of the Windows command line interface and the DIVArchive Control GUI. Inserting the tape is an optional part of the workflow but it is necessary to access the objects on the tape. It is possible to run the **`importTape`** command line utility to enter the tape's metadata into the DIVArchive Database, and still keep the tape externalized. However, in order to access the objects on the tape, the tape must be inserted via the DIVArchive **`insert tape`** function.

The following procedure is used for importing tapes into the DIVArchive System.

1. Start the Windows command line interface by clicking on the **Start Menu**, then click on **Run**. Type **`cmd`** in the text box and click the **OK** button. The Windows command line interface will start.
2. Copy the exported XML and FFM Files into the **`DIVA_HOME\Manager\bin`** folder
3. Change to the **`DIVA_HOME\Manager\bin`** folder:

```
C:\ cd DIVA_HOME\Manager\bin
```

Figure 4: Windows Command Interface



4. Run the **importTape** command using any command line options required (*refer to the table below for valid options*).

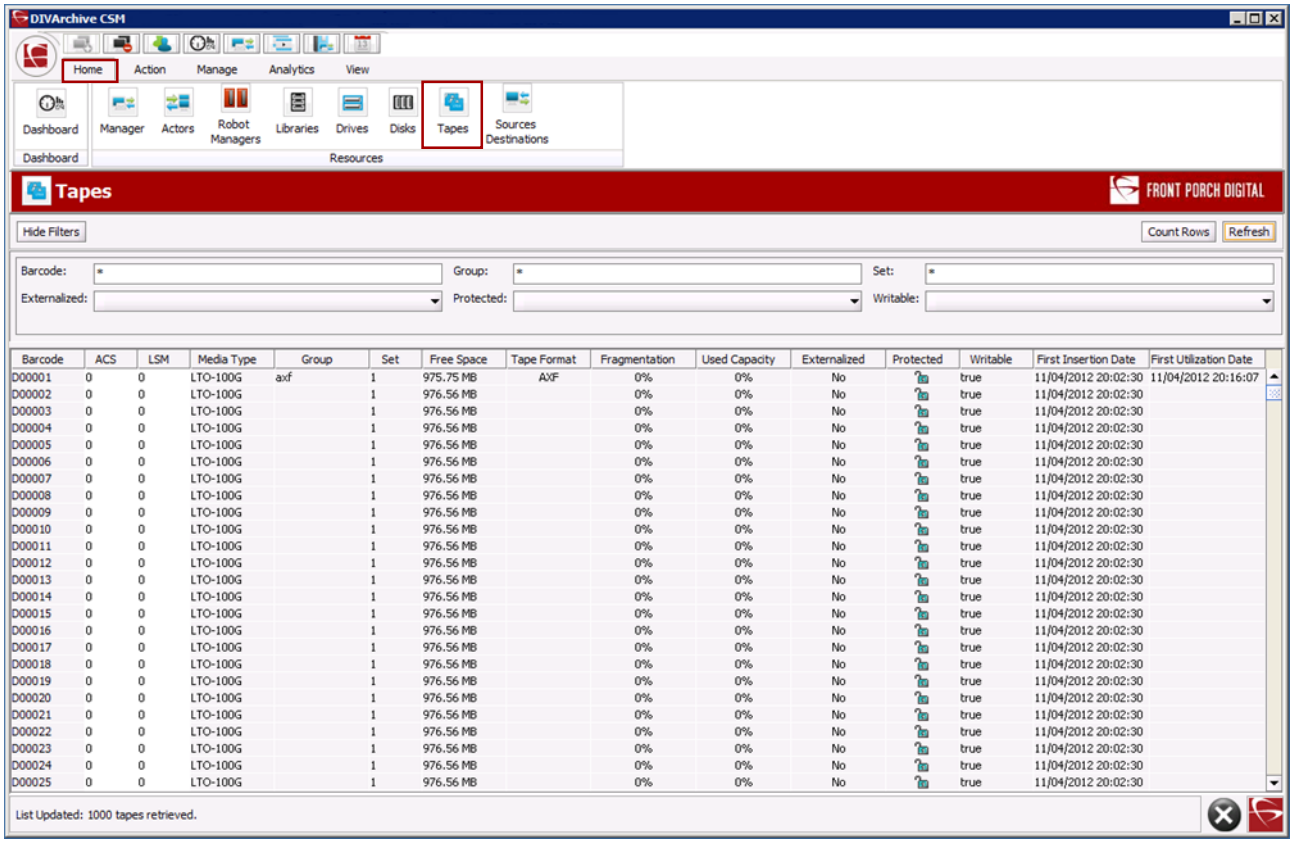
*Table 4: **ImportTape** Command Line Options*

Option	Definition
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(s) containing exported Tape Metadata or a folder that contains these files.
-skipIfNameExists	Skip import of objects with naming conflicts. **Note: The default behavior is that if the Object Name and Category already exist, the utility will abort 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 DIVArchive Database. The Tape Object must have the same Object Name/Category, Components, and Checksums as the object in the database.

Option	Definition
`-useImportDateAsArchiveDate`	Changes the imported object's original archive date to the date of import on the system where the objects are being imported. This does not change the original archive date in the exported XML file or on the original system where the object was exported from; only on the system where the object was imported.

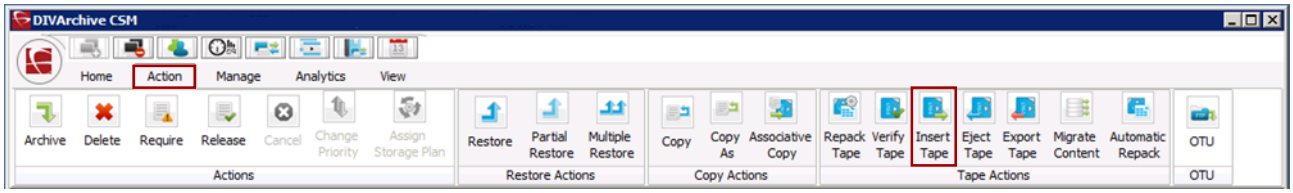
In the DIVArchive Control GUI, click on the **Home Tab** and then the **Tapes Button** to show the list of tapes identified in the system via the **Tapes Panel** as shown in the figure below. Imported tapes may be kept externalized, however in order to restore the objects on a tape it must be inserted into the library.

Figure 5: Control GUI – Tapes View



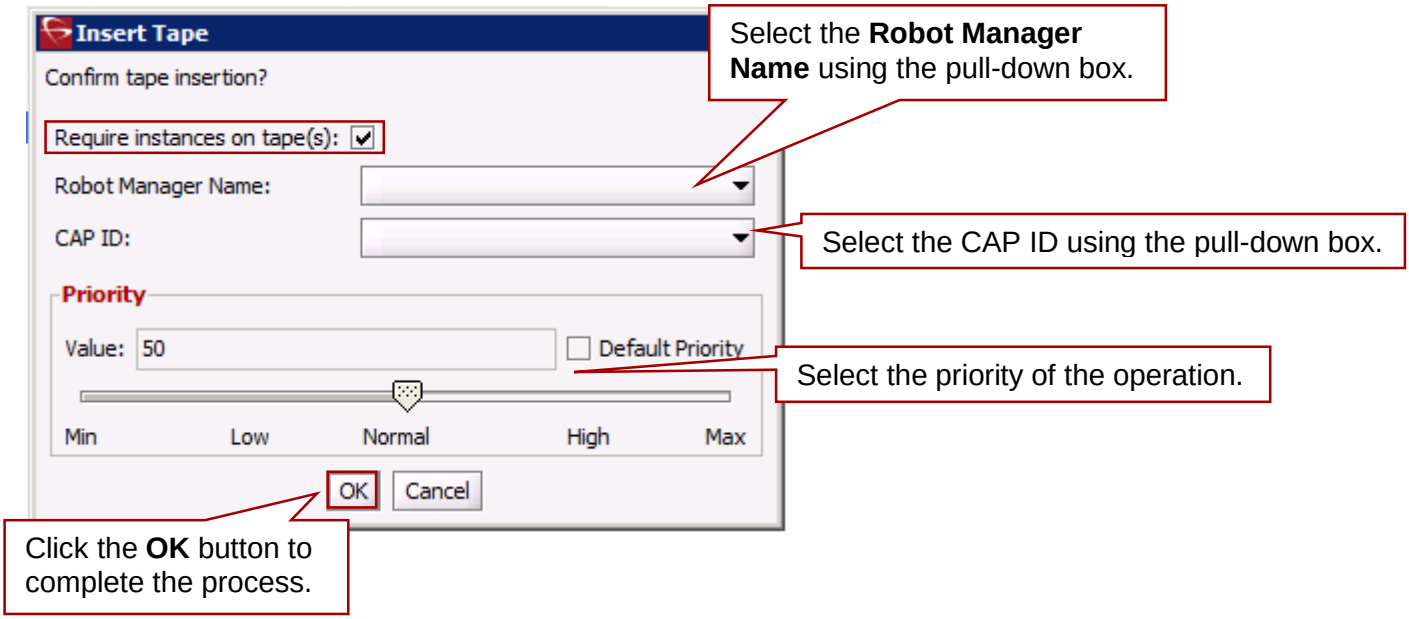
5. Highlight the desired tape(s) and then navigate to the **Action Tab** on the Ribbon Bar and click on the **Insert Tape Button** to open the **Insert Tape Dialog Window**.

Figure 6: Control GUI Action Tab Showing the Insert Tape Button



6. The **Insert Tape Dialog Window** will appear as shown in the figure below.

Figure 7: Insert Tape Popup Window



As displayed in the figure above:

a. If the object's instance needs to pre-exist in the database before the tape is inserted, check the box next to **Require instances on tape(s)**; otherwise leave it unchecked.

b. Select the appropriate **Robot Manager Name** using the pull-down box.

c. Select the appropriate **CAP ID** using the pull-down box.

d. Select the priority value for the insert operation.

e. Once the tape(s) is inserted, restoration of the objects on the imported tape(s) will be possible.

3 Troubleshooting

3.1 Export failed with the following sequence of events:

```
Robot Manager Error : Error while ejecting
tapes: StatusCode[70:INTERNAL_ERROR]
Request step is STEP_WAITING_FOR_OPERATOR()
```

Resolution:

First check if the CAP that 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 particularly an issue if using a set 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.

3.2 I received an error similar to this during Export:

```
Invalid parameter : Tape Y00105 must be included into export list
```

Resolution:

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 in order for the export to succeed.

3.3 I received the following error when importing:

```
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 you are importing already exists in the DIVArchive System. It is likely that the Tape Metadata for the tape you wish to import already exists in the DIVArchive Database, and you need only to perform an Insert Tape operation to use the tape. Verify that the tape contains the correct objects by using the DIVArchive Control GUI.

3.4 When Importing, I encounter an error similar to:

```
The following errors were found in tapeset-[Y00109].xml\Tape Y00109
has unsupported type 19.
```

Resolution:

The **type** refers to the **`mediaTypeId`**. The **Media Type ID** is an ID that represents the type of tape media being exported. DIVArchive exports a **`mediaTypeId`** field, which corresponds to the Id column in the Tape Properties table under the **Tapes** tab in the DIVArchive Configuration Utility. 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. Insure that the block size and total size of the **`mediaType`** in the source DIVArchive System matches the **`mediaType`** definition in the destination.

3.5 Why did the import process abort without importing anything?

Resolution:

There are several reasons why the import process may abort without completing successfully:

- 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 abort and an error will be written to the log file.
- If the Object Name and Category already exist, and the **`–skipIfNameExists`** or **`–addAsInstanceIfNameExists`** options are not passed, the utility will abort without importing the tape(s).
- If the Manager detects a database error, the import process will be aborted and any operations performed during the failed import will be rolled back and not saved in the system.

4 Frequently Asked Questions

4.1 What is the export XML and FFM file compatibility?

The exported XML and FFM files, when generated, will be able to be imported into the version of DIVArchive that it was exported from, and future versions of DIVArchive. DIVArchive allows more than one set of tapes (*spanned or not*) to be exported to and imported from a single file.

Note: Exported Metadata from the DIVArchive 7.3 Export function cannot be imported into DIVArchive versions prior to version 7.0. However, exported Metadata created from versions of DIVArchive prior to 7.3 can be imported into the DIVArchive 7.3 System.

4.2 What is a Media Type ID?

The **Media Type ID** is a proprietary DIVArchive identifier that represents the type of tape media being exported. DIVArchive exports a **`mediaTypeId`** field, which corresponds to the Id column in the Tape Properties table under the **Tapes** tab in the DIVArchive Configuration Utility. 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 insure that the block size and total size of the **`mediaType`** in the source DIVArchive System matches the **`mediaType`** definition in the destination. This becomes especially important if the tape is ever repacked.

4.3 What are the unsupported DIVArchive attributes?

The **`markedAsDeleted`** is an internal attribute and is not exported or imported via the Export/Import Utility. In addition, the state of checksum verification (*verified, partially verified, etc.*) 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 DIVArchive request.

APPENDIX

A1 Non-Spanning Export XML Sample

```
<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>
```

A2 Spanning Export XML Sample

```
<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>
```