

Oracle® DIVArchive

Drop Folder Monitor (DFM) User's Guide

Release 7.5

E79770-01

November 2016

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Lou Bonaventura

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

1	INTRODUCTION.....	1
1.1	SYSTEM OVERVIEW.....	1
1.2	DOCUMENT CONVENTIONS	1
1.3	DEFINITIONS, ACRONYMS, AND SPECIAL TERMS.....	2
1.4	OVERVIEW.....	3
1.5	DROP FOLDER MONITOR (DFM) FEATURES.....	6
1.5.1	Single File Mode Drop Folder.....	6
1.5.2	File Set Drop Folder.....	7
2	DFM SOLUTIONS	8
2.1	BASIC DFM SOLUTION FOR SINGLE MODE FOLDER.....	8
2.1.1	Configuration of Basic DFM Solution.....	9
2.2	DFM SOLUTION FOR FILE SET FOLDERS ON LOCAL DISK AND FTP SERVERS	10
2.2.1	Recursive Archive.....	11
2.2.2	Configuration of DFM Solution for File Set Folders.....	14
2.3	DFM SOLUTION FOR DELETE MODE.....	15
2.3.1	Configuration of DFM Solution for Delete Mode.....	15
2.4	DFM SOLUTION FOR DIVANET	16
2.4.1	Configuration of DFM Solution for DIVAnet.....	17
3	DFM KEY FEATURES AND CONFIGURATION OPTIONS.....	19
3.1	DFM MONITORING WORKFLOW AND ADDITIONAL CONFIGURATION	19
3.1.1	Overview and Definitions	19
3.1.2	Algorithm and Workflow for Single File Mode Folders.....	19
3.1.3	Algorithm and Workflow for File Set Mode Folders.....	21
3.1.4	Incomplete File Threshold (<incompleteThreshold>).....	22
3.1.5	Processing of Incomplete Files (<incompleteProcessingStrategy>).....	23
3.1.6	MDF Configuration Priority (<mdfConfigPriority>).....	23
3.2	ADDITIONAL CONFIGURATION FOR DIVARCHIVE SYSTEM OPERATIONS	24
3.2.1	Delete Before Archive (<deleteBeforeArchive>).....	24
3.2.2	Number of Files to Archive Wildcard (<numberOfFilesToArchiveWildcard>).....	24
3.2.3	Object Name Conversion Rules (<objectNameConvertRule>).....	24
3.2.4	Original Server (<originalServer>).....	25
3.2.5	Original Path (<originalPath>).....	26
3.2.6	DIVArchive Media Name (<divaMediaName> and <divaMediaNamePattern>)	26
3.3	ADDITIONAL CONFIGURATION FOR THE MONITORED FOLDER	26

3.3.1	MDF File Extension (<mdfExtension>)	26
3.3.2	Source/Destination DIVArchive Path (<sourceDestinationDIVAPath>)	27
3.3.3	Archive File Path Template (<archiveFilePathTemplate>)	27
3.3.4	Archive Filename Template (<archiveFileNameTemplate>)	28
3.3.5	Source/Destination Path Samples for Different Configurations	28
3.3.6	File Filter (<fileFilter>)	30
3.3.7	Delete Parent Directory and Content Directories (<deleteParentDirectoryAndContentDirectories>)	30
4	FOLDER CONFIGURATION EXAMPLES	32
4.1	EXAMPLE OF A LOCAL FOLDER IN FILE SET MODE	32
4.2	EXAMPLE OF AN FTP FOLDER IN SINGLE FILE MODE	33
4.3	EXAMPLE OF A CIFS FOLDER	34
4.4	TRACING CONFIGURATION	35
5	ADVANCED DFM CONFIGURATION	37
5.1	DFM FULL CONFIGURATION FILE	37
5.2	GLOBAL DFM CONFIGURATION	38
5.3	FOLDERS DEFAULT CONFIGURATION	40
5.4	FOLDER-SPECIFIC CONFIGURATION	43
5.5	ADVANCED CONFIGURATION OF FILE ROOT PATH IN SINGLE FILE MODE	46
6	DFM OPERATION, ADMINISTRATION AND MONITORING	50
6.1	OVERVIEW OF STARTING, STOPPING AND RESTARTING DFM	50
6.1.1	Starting and Restarting DFM	50
6.1.2	Running DFM	50
6.1.3	Stopping DFM	50
6.2	SUMMARY OF DFM ADMINISTRATION	51
6.3	DIVARCHIVE DFM COMMAND LINE INTERFACE	51
6.4	MONITORING OF THE DIVANET ACCESS GATEWAY VIA THE DIVARCHIVE CONTROL GUI	53
6.5	MONITORING OF DFM USING LOGS	54
6.6	RUNNING MULTIPLE INSTANCES OF DFM ON THE SAME MACHINE	55
APPENDIX		56
A1	DIVARCHIVE OPTIONS AND LICENSING	56
A2	OBJECT CONFIGURATION	57
A3	METADATA FILE	58
A4	SAMPLE OF THE SAMPLE.MDF FILE IN THE OLD MDF FORMAT	59
A5	EXAMPLE OF THE SAMPLE.XML IN XML FORMAT	60
A6	SAMPLE OF THE FULL TRACE LOG FILE	61
A7	SAMPLE OF FULL DFM CONFIGURATION FILE (DFM.CONF.INI)	65

Tables Index

Table 1: System Component Icons for Non-Screenshot Drawings	2
Table 2: Definitions, Acronyms, and Special Terms	2
Table 3: Global DFM Configuration Parameters.....	38
Table 4: Folders Default Configuration Parameters.....	40
Table 5: Folder Specific Configuration Parameters	43
Table 6: Object Configuration Parameters	57

Figures Index

Figure 1: Basic DFM Workflow for Single Mode Folder	8
Figure 2: Basic DFM Workflow for Set Mode Folder.....	11
Figure 3: Request Properties – Using the –r Option	12
Figure 4: Configuration Utility - Source/Destination Configuration using the –r Option	12
Figure 5: DFM Solution in Delete Mode Workflow	15
Figure 6: DFM Solution for DIVAnet Workflow	16
Figure 7: DIVArchive Control GUI Showing Status DFM Archive Requests.....	53

1 Introduction

1.1 System Overview

Oracle DIVArchive 7.5 enables support for Oracle Linux 7 x86_64 and later for all core components. Linux-based Actors have some limitations associated with them, and Windows-based Actors must be used for the following:

- Avid Connectivity support
 - See *A1 DIVArchive Options and Licensing for DIVArchive licensing information*.
- Transcoding operations
- Tape Reading Utility

Additionally, long path names are now supported on both Windows and Linux.

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

See the *Oracle DIVArchive Installation and Configuration Guide*, and the *Oracle DIVArchive Supported Environments Guide* in the *Oracle DIVArchive 7.5 Core Documentation Library* for information about running Windows services, Linux services, transcoders, and certain limitations when running in the Linux environment.

See A1 DIVArchive Options and Licensing for DIVArchive licensing information.

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 Oracle 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 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.

Table 1: System Component Icons for Non-Screenshot Drawings

Icon	Component	Icon	Component
	Oracle DIVArchive System		FTP Server
	Storage		
	Software Application		Objects
			Folders

1.3 Definitions, Acronyms, and Special Terms

Table 2: Definitions, Acronyms, and Special Terms

Term	Definition
CIFS	Common Internet File System: Microsoft file sharing protocol. Note: Linux-based Actors do not support UNC paths for CIFS sources and destinations.
Complex Object	An Object is defined as a Complex Object when it contains 1,000 (<i>configurable</i>) or more components. Complex Object handling may differ from non-Complex Objects as noted throughout this document.
Component	A file that is part of a DIVArchive Object.
DFM	Drop Folder Monitor.
Drop Folder	Folder on a local disk, FTP Server, or a CIFS shared folder designated for Single File mode and/or File Set storage that will be monitored by DFM and from which files will be archived.
FOIB	File Object Information Base: Persistent storage used by DFM to track its processing state.

Term	Definition
FTP	<p>File Transfer Protocol: used for transferring files from one machine to another.</p> <p>Note: When the DFM Drop Folder is configured on an FTP Server, all content (<i>files and folders expected to be archived</i>) should be <u>copied to DFM Drop Folder via FTP</u>. If the content is copied to the DFM Drop Folder by other means (<i>local copy via OS, remote desktop copy, etc.</i>) there is no guarantee of proper processing by DFM.</p> <p>When DFM is used in a Linux environment to monitor an FTP folder, it must be configured as in the following example:</p> <p>User: diva</p> <p>User Home Directory: /ifs</p> <p>Folder to be Monitored: /ifs/folder1</p> <p>Correct DFM Configuration: ftp://diva:password@host_ip/folder1</p> <p>Incorrect DFM Configuration: ftp://diva:password@host_ip/ifs/folder1</p>
MDF	Metadata File extension (<i>.mdf</i>)
Metadata File	A file containing information about the files included in a File Set.
DFM Incomplete Files	Files that were put into the Drop Folder without the Metadata File (<i>for File Set folders</i>) and files that cannot be archived after a specific number of attempts.

1.4 Overview

Drop Folder Monitoring (DFM) allows users and 3rd-party applications to deliver content to be archived by copying related file(s) to a folder, an FTP server, or CIFS.

Note: Linux-based Actors do not support UNC paths for CIFS sources and destinations.

With DFM there is no need for an application or user to send the archive command. Each time a new file, or set of files (*File Set*), is placed in a monitored Drop Folder DIVArchive automatically archives the file(s) and creates the related DIVArchive Object according to the rules set in the configuration file for that particular Drop Folder.

The flexibility of DFM allows for a myriad of possible configuration combinations. To make configuration and operations easier for the end user, Oracle has combined the most commonly used configurations into four basic DFM Solutions. If necessary, advanced DFM configuration can be utilized to employ DFM well beyond the four solutions described in this guide.

DFM supports a number of automated operations. Example configurations have been included to aid in the configuration of specific workflows:

- **dfm.conf.delete.ini** contains the minimum configuration parameters to configure DFM for use as a File Deletion Solution.

- `dfm.conf.single.local.ini` is a template for configuring DFM for automated local, Single File-based Archive Operations.
- `dfm.conf.ini` is an example template demonstrating how multiple workflows and Drop Folders can coexist in a single configuration file, and are modular by design.

Oracle offers the four most common DFM Solutions as described in the following sections. Select the solution, or solutions, that meet your requirements and follow the configuration instructions for those particular solutions.

Section 2.1: Basic DFM Solution for Single Mode Folders

This DFM Solution is capable of monitoring a local disk, FTP transfers, and CIFS folders. Each file found in the Drop Folder is archived by the DIVArchive System as a single object. If your requirements include only single files then this solution will work perfectly.

- Supported objects to be archived:
 - Single file only.
- Supported major configuration parameters:
 - Oracle DIVArchive Manager configuration
 - Drop Folder URL.
 - Category Name of objects to be archived.
 - Source/Destination used in archive request.
 - Templates for generation of File Path Root and File Name.
 - DIVArchive Media Name used in archive request.

Notes:

If the connection to the disk, FTP or CIFS is unavailable (or *disconnected*) for a period of time (when the *fileReadyThreshold* time has elapsed) the file will be considered “complete” when it is actually incomplete.

The *fileReadyThreshold* parameter is only found in the `dfm.conf.ini` and not in the `dfm.single.conf.ini` configuration file. However, this parameter may be manually added to the `dfm.single.conf.ini` file if desired/required.

Linux-based Actors do not support UNC paths for CIFS sources and destinations.

2.2: DFM Solution for File Set Mode Folders

This DFM Solution offers more flexibility than the Basic Solution. This configuration is capable of monitoring the local disk, FTP Server, and Non-secured CIFS folders that contain either Single Files or File Sets (*Secured CIFS can also be used if the service is run under a suitable user account*).

Note: Linux-based Actors do not support UNC paths for CIFS sources and destinations.

File Sets include a Metadata File that contains the details of the files included in the set. Each Metadata File found in the Drop Folder is parsed and a new object (*usually with more than one component*) is archived in the DIVArchive System.

- Supported objects to be archived:
 - Single and multiple components (*files*).
 - Recursive Archive (*folders and files*).
- Supported major configuration parameters:
 - Manager configuration.
 - Drop Folder URL.
 - DIVArchive Media Name used in archive request.

2.3: DFM Solution for Delete Mode

This DFM Solution monitors only the local disk and removes files that are obsolete. If a subfolder is empty for more than the configured allowed time period, it will also be removed. This solution is used in conjunction with the other DFM Solutions and can be included in all DFM Systems.

- Supported major configuration parameters:
 - Folder URL.
 - Interval that specifies when the file will be deleted from folder.

2.4: DFM Solution for Oracle DIVAnet

This DFM Solution monitors only the local disk and is capable of using either Single Files or File Sets. Each Metadata File found in the Drop Folder is parsed and a new object (*usually with more than one component*) is archived in the DIVArchive System.

See A1 *DIVArchive Options and Licensing for DIVArchive licensing information.*

This solution is used with Oracle DIVAnet systems that are configured with inter-site transfer capabilities. Contact your Oracle Sales Support Specialist for detailed information on various DIVAnet Solutions available.

- Supported objects to be archived:
 - Single and multiple components (*files*).
 - Recursive Archive (*folder and files*).
- Supported major configuration parameters:
 - Manager configuration.
 - Drop Folder URL.
 - DIVArchive Media Name Pattern for generation of the media name used in the archive request.
 - Source/Destination Name
 - Source/Destination Path

1.5 Drop Folder Monitor (DFM) Features

Drop Folder Monitor is a component of DIVArchive that can be installed with the DIVArchive installation package.

DFM monitors designated Drop Folders and requests DIVArchive to archive objects that are found in these folders. Drop Folder Monitor can monitor up to 20 local folders, FTP folders, and CIFS folders each with its own individual configuration.

Note: Linux-based Actors do not support UNC paths for CIFS sources and destinations.

For example, if the archive contains 2 media groups, News and Movies, then 2 Drop Folders should be configured, 1 for News and 1 for Movies. However, if **File Set Mode** is used (*or an advanced configuration setup*), a single Drop Folder may be utilized instead.

Once archived the files are deleted from the folder they have been archived from.

There are 2 types of Drop Folders:

- Single File Mode
- File Set Mode

Notes:

- **Best practices dictate that files should be copied to the Drop Folder and not moved. After the configured time period, if the file is considered incomplete, it may be deleted. Copying the file rather than moving it will ensure a backup copy is available in the event of an unwanted deletion.**
- **When the DFM Drop Folder is configured on an FTP Server, all content (*files and folders expected to be archived*) should be copied to DFM Drop Folder via FTP. If the content is copied to the DFM Drop Folder by other means (*local copy via OS, remote desktop copy, etc.*) there is no guarantee of proper processing by DFM.**

1.5.1 Single File Mode Drop Folder

A **Single File Mode Drop Folder** is used to automatically archive objects that are made up of a single file (*e.g. convenient for **MXF**, **GXF**, etc. with single file assets*). There can be many single files in the folder but each will only be archived as a Single File Object.

The Object Name is the filename including the extension. The Object Category, priority and other archive command parameters are defined in a Drop Folder configuration file (*dfm.conf*) which contains the description of all of the folders to be monitored by the DFM Module.

When a file appears in a configured Drop Folder, DFM monitors its size and if the file size has not changed after a pre-configured time threshold (*last modification time*), then the file is considered complete, and the archive is initiated. Once the file is archived it is automatically deleted from the Drop Folder.

1.5.2 File Set Drop Folder

A **File Set Drop Folder** is used to automatically archive objects that contain more than one file and is compatible with Complex Objects.

When monitoring a File Set Drop Folder, DFM checks for the presence of a Metadata File (`.mdf`), which describes the details of the content to be archived, in order to know when the content to be archived is ready. The archive will only start after this file is available to DFM.

Using the Metadata File allows for specifying the Object Name, Category and user comments on a per object basis.

First, either the user or application must create a folder in the Drop Folder for the set of files intended to be archived as a single object in DIVArchive. This folder must have the `-allow_delete_on_source` and `-r` options configured.

Once the folder is created, the data file(s) have to be copied to the folder.

WARNING: The last file to be created in the File Set Drop Folder must be the Metadata File (`.mdf`) which then triggers the archive operation. If the Metadata File is transferred BEFORE the actual data files, content loss can occur!

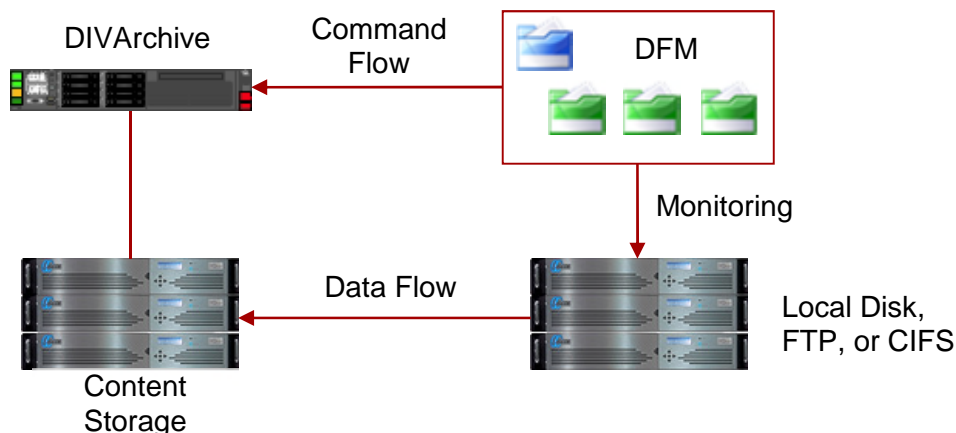
Once DFM has opened and parsed the Metadata File, the object archiving process is initiated. After the object is successfully archived, DFM deletes the folder and all of the files within it including the `.mdf` file.

2 DFM Solutions

2.1 Basic DFM Solution for Single Mode Folder

This DFM Solution is capable of monitoring a local disk, FTP transfers, and CIFS folders. Each file found in the Drop Folder is archived by the DIVArchive System as a single object. As files are dropped into the folder they are archived as single files into DIVArchive. Multiple files can be in the Drop Folder at the same time, but they will be handled by the DFM one at a time.

Figure 1: Basic DFM Workflow for Single Mode Folder



- Supported Source/Destinations:
 - Local disks
 - FTP folders
 - **When the DFM Drop Folder is configured on an FTP Server, all content (*files and folders expected to be archived*) should be copied to DFM Drop Folder via FTP. If the content is copied to the DFM Drop Folder by other means (*local copy via OS, remote desktop copy, etc.*) there is no guarantee of proper processing by DFM.**
 - CIFS folders
 - Linux-based Actors do not support UNC paths for CIFS sources and destinations.
- Supported objects to be archived:
 - Single component (*file*) only.
- Supported major configuration parameters:
 - Manager configuration.
 - Drop Folder URL.
 - Category Name of objects to be archived.
 - Source/Destination used in archive request.
 - Templates for generation of File Path Root and File Name.
 - DIVArchive Media Name used in archive request.

2.1.1 Configuration of Basic DFM Solution

In the basic DFM solution, each file found in the Drop Folder is archived by the DIVArchive System as single object.

Use the following procedure to configure the DFM Solution for a **Single Mode Folder**. Editing the DFM configuration file is necessary to make the following changes.

1. Rename the `dfm.conf.single.local.ini` file to `dfm.conf`.

`DIVA_HOME\Program\conf\dfm\dfm.conf`

2. Specify the host and port using `<managerConnetion>`.

Example:

```
<managerConnetion>
  <address host="localhost" port="9000"/>
</managerConnetion>
```

3. Specify the Windows Service Name for `<serviceName>`.

- This parameter should be written on one line and there should not be any symbols or empty spaces at the beginning of the line.

Example:

```
<serviceName>DIVArchiveDFM</serviceName>
```

4. Specify the folder URL using `<url>`.

Example:

```
<url>file:///c:\DROPFOLDER\</url>
```

- Refer to Folder Configuration Examples for folder configuration examples.

5. Specify the Category Name of objects to be archived using `<categoryName>`.

Example:

```
<categoryName>Category</categoryName>
```

6. Specify the Source/Destinations to be used in archive requests using `<sourceDestinationDIVAName>`.

Example:

```
<sourceDestinationDIVAName>DISK</sourceDestinationDIVAName>
```

7. Specify the file path to the folder containing the files that DIVArchive will archive using `<archiveFilePathTemplate>`.

Example:

```
<archiveFilePathTemplate platform="DETECT" options="">
  URL_TO_FILE
</archiveFilePathTemplate>
```

- Specify the template for the generation of the File Name using `<archiveFileNameTemplate>`.

Example:

```
<archiveFileNameTemplate platform="DETECT" options="">
    Filename.Ext
</archiveFileNameTemplate>
```

- Specify the DIVArchive Media Name used in the archive request using `<divaMediaName>`.

Example:

```
<divaMediaName>Array1</divaMediaName>
```

- Rename the `DIVA_HOME\Program\conf\dfm\dfm.trace.ini` file to `dfm.trace`.

- If this step is not completed the installation will fail because it cannot find the trace file.**

- Install the DFM Service:

```
DIVA\Program\InterLink\dfm\bin\dfm.bat install
```

- Start the DFM Service:

- `DIVA\Program\InterLink\dfm\bin\dfm.bat start`
- Alternatively, proceed to the Windows Services Control Panel and start the service from there.

2.2 DFM Solution for File Set Folders on Local Disk and FTP Servers

This DFM Solution offers more flexibility than the Basic Solution. This configuration is capable of monitoring the local disk, FTP Server, and CIFS folders that contain either Single Files or File Sets.

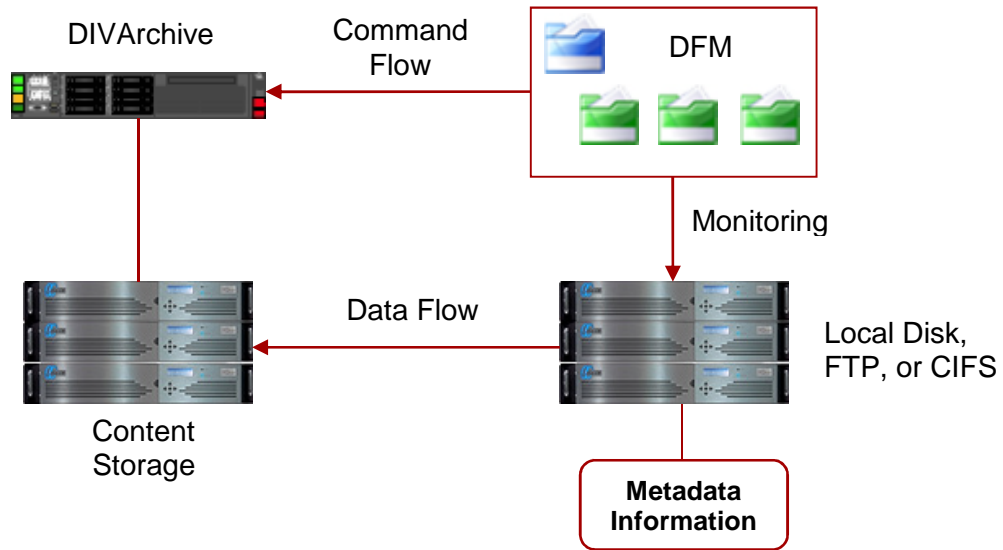
DFM supports **Single** and **Set** based folder types. Each file that is dropped into a Single-type folder results in separate archive per file. Set-type folders, on the other hand, provide a means of archiving objects comprised of multiple files. Set-mode archive operations require the use of Metadata Files, which contain the details of the file-set comprising a specific archive operation (see A3 *Metadata File*). Files for separate archive operations can coexist in a set-mode folder, as the scope of files included in each archive operation is determined by the file list in each MDF file placed in the set-mode folder.

AXF format files are, at the filesystem level, single files, but contain many files internally. AXF files can be archived into DIVArchive, however this requires the use of a set-mode folder and an MDF File; in this case, the MDF File contains the list of files contained within the AXF file, as well as the target parent attributes of the object to be used when archived.

WARNING: The last file to be created in the File Set Drop Folder must be the Metadata File (.mdf) which then triggers the archive operation. If the Metadata File is transferred BEFORE the actual data files, content loss can occur!

For samples of Metadata Files in DIVArchive DFM format refer to Appendixes A1 through A4.

Figure 2: Basic DFM Workflow for Set Mode Folder



- Supported Source/Destinations:
 - Local disks.
 - Non-secured CIFS.
 - Secured CIFS is possible if the Windows DFM Service is run under a suitable user account.
 - FTP Servers.
 - **When a DFM Drop Folder is configured on an FTP Server, all content (*files and folders expected to be archived*) should be copied to the DFM Drop Folder via FTP. If the content is copied to the DFM Drop Folder by other means (*local copy via OS, remote desktop copy, etc.*) there is no guarantee of proper processing by DFM.**
- Supported objects to be archived:
 - Single and multiple components (*files*).
 - Recursive Archive (*folders and files*).
- Supported major configuration parameters:
 - Manager configuration.
 - Drop Folder URL.
 - DIVArchive Media Name used in archive request.

2.2.1 Recursive Archive

DFM Recursive Archive is enabled when the DFM folder is configured to use File Set Mode (*folder type is set*). In this mode the Archive File Set will be defined in the specific object configuration file (*.mdf file*).

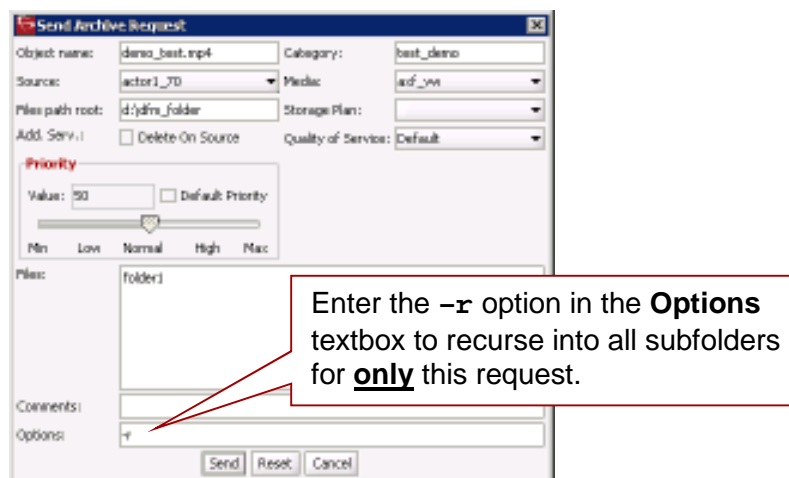
This feature defines a special character - asterisk (*) - which can be used in the DFM Metadata File (*.mdf file*). Applying asterisk (*) to the end of folder path will cause all files under this path to be archived, but not subfolders.

Use of the `-r` (*recursive*) option in the configuration of the drop folder achieves different results than using the “*” option. When the `-r` option is included in the request the Oracle DIVArchive Actor will recurse into all subfolders whether the “*” is used or not. Conversely if the `-r` option is not used, the Actor will only select the files in the immediate folder. In this case, use of the “*” parameter will still cause archiving of all files in the immediate folder action to occur. When using this option, be aware of the Request and Source/Destination configurations to achieve the desired results.

There are two locations where the `-r` option may be configured:

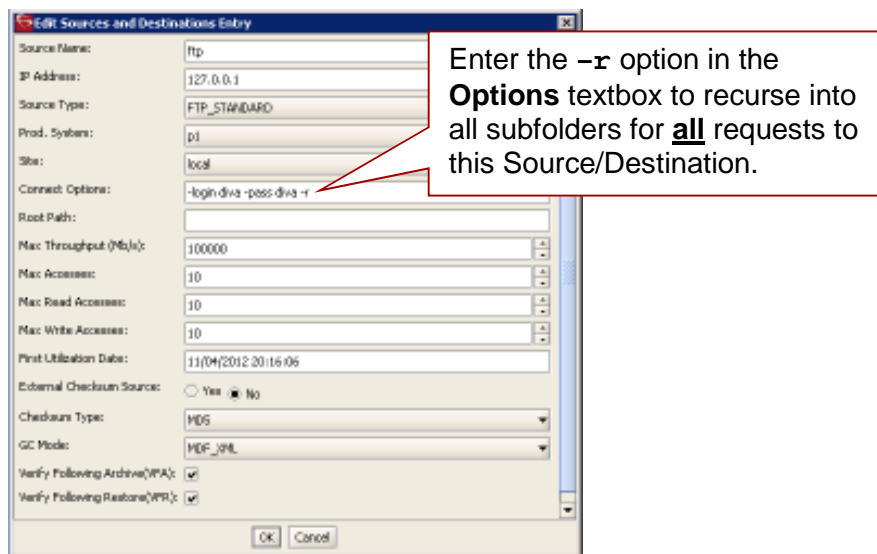
1. In the Request Options when creating the request (see the figure below).
 - o Applicable **only** to the current request.

Figure 3: Request Properties – Using the `-r` Option



2. In the Source/Destination options.
 - o Applicable to **all** requests to this Source/Destination (see figure below).

Figure 4: Configuration Utility - Source/Destination Configuration using the `-r` Option



Below are several simple examples (*excerpts from the .mdf file*), which enables recursive archive for the specific folders and files:

Example:

```
<fileList>
  folder1/*
  test/folder2/*
  folder3/test.txt
</fileList>
```

Using this configuration:

- Every file under `folder1` and `test/folder2` will be archived (*but not subfolders*).
 - If the `-r` option was used in the **Request Options**, or on the **Source/Destination Configuration**, all subfolders and their files will also be archived.
- For the `folder3/test.txt`, only the `test.txt` file located in `folder3` will be included.

Example:

```
<fileList>
  *
</fileList>
```

Using this configuration:

- Every file in the specified folder will be archived (*but not subfolders*).
- If the `-r` option was used in the **Request Options**, or on the **Source/Destination Configuration**, all subfolders and their files will also be archived.

Example:

```
<fileList>
  folder1/*
</fileList>
```

Using this configuration:

- Every file under `folder1` will be archived (*but not subfolders*).
- If the `-r` option was used in the **Request Options**, or on the **Source/Destination Configuration**, all `folder1` subfolders and their files will also be archived.

Note: Recursive Archive does take into consideration the use of the `-file_order FILE_FIRST` and `-file_order DIRS_FIRST` command options.

2.2.2 Configuration of DFM Solution for File Set Folders

Use the following procedure to configure the DFM Solution for a **File Set Folder**, Default Filter for Local Disks, and FTP.

1. Rename the `dfm.conf.set.local.ftp.ini` file to `dfm.conf`.
`DIVA\Program\conf\dfm\dfm.conf`

2. Specify the host and port using `<managerConnetion>`.

Example:

```
<managerConnetion>
  <address host="localhost" port="9000"/>
</managerConnetion>
```

3. Specify the Windows Service Name using `<serviceName>`.

- This parameter should be written on one line and there should not be any symbols or empty spaces at the beginning of the line.

Example:

```
<serviceName>DIVArchiveDFM</serviceName>
```

4. Specify the `<folderConfig>` node using `<type>`.

- The folder should be File Set and therefore the **set** value should be specified as in the example below.

Example:

```
<type>set</type>
```

5. Specify the folder URL using `<url>`.

- Refer to Folder Configuration Examples for folder configuration examples.

Example:

```
<url>ftp://diva:diva@localhost:21/dropfolder</url>
```

6. Specify the DIVArchive Media Name used in archive request using `<divaMediaName>`.

Example:

```
<divaMediaName>Array1</divaMediaName>
```

7. Rename the `DIVA_HOME\Program\conf\dfm\dfm.trace.ini` file to `dfm.trace`.

- **If this step is not completed the installation will fail because it cannot find the trace file.**

8. Install the DFM Service:

```
DIVA\Program\InterLink\dfm\bin\dfm.bat install
```

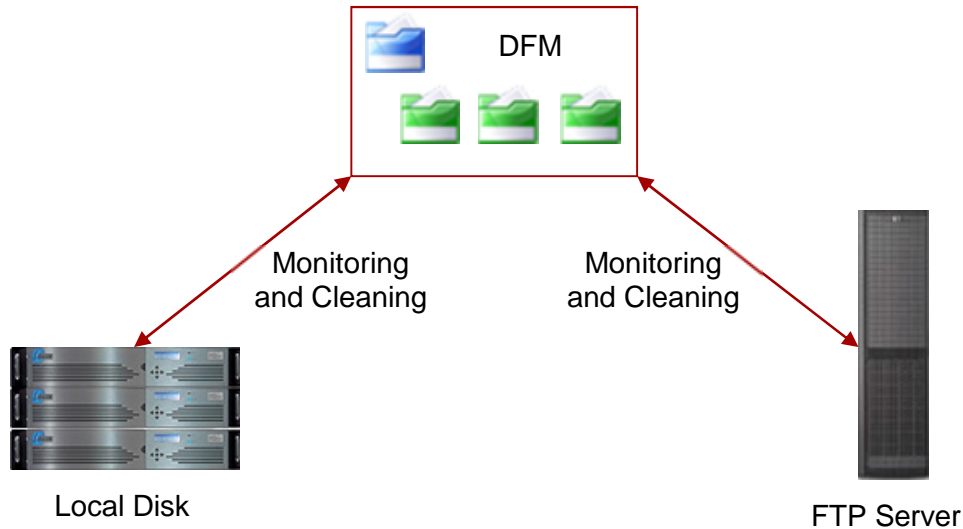
9. Start the DFM Service:

```
DIVA\Program\InterLink\dfm\bin\dfm.bat start
```

2.3 DFM Solution for Delete Mode

This DFM Solution monitors the specified folder and removes files that are obsolete. If a subfolder is empty for more than the configured allowed time period, it will also be removed. This solution is used in conjunction with the other DFM Solutions and should be included in all DFM Systems.

Figure 5: DFM Solution in Delete Mode Workflow



- Supported major configuration parameters:
 - Folder URL
 - Interval that specifies when the file will be deleted from folder.

2.3.1 Configuration of DFM Solution for Delete Mode

Use the following procedure to configure the DFM Solution for Delete Mode.

1. Rename the `dfm.conf.delete.ini` file to `dfm.conf`.
`DIVA\Program\conf\dfm\dfm.conf`
2. Specify the interval that will identify when the file will be deleted from the folder using `<fileDeleteThreshold>`.
 - File Delete Threshold in seconds (for folders in *DELETE* mode only).
 - This interval will specify when a file will be deleted from a DFM folder after copying the file to the folder is completed.
 - The state of completion will be identified by the file size. As the file is copied, the size of the file will continue to grow until the copy is complete. Once the file size no longer increases, the system will recognize the file copy operation as being completed.
 - It should be a positive integer value.
 - Default of 86400 seconds equals 24 hours.

Example:

```
<fileDeleteThreshold>86400</fileDeleteThreshold>
```

3. Specify the Windows Service Name using `<serviceName>`.

Example:

```
<serviceName>DIVArchiveDFM</serviceName>
```

4. Specify the `<folderConfig>` node using `<mode>`.

Example:

```
<mode>Delete</mode>
```

5. Specify the folder URL using `<url>`.

Example:

```
<url>file:///c:/CLEAN_FOLDER</url>
```

- Refer to Folder Configuration Examples for folder configuration examples.

6. Rename the `DIVA_HOME\Program\conf\dfm\dfm.trace.ini` file to `dfm.trace`.

- **If this step is not completed the installation will fail because it cannot find the trace file.**

7. Install the DFM Service:

```
DIVA\Program\InterLink\dfm\bin\dfm.bat install
```

8. Start the DFM Service:

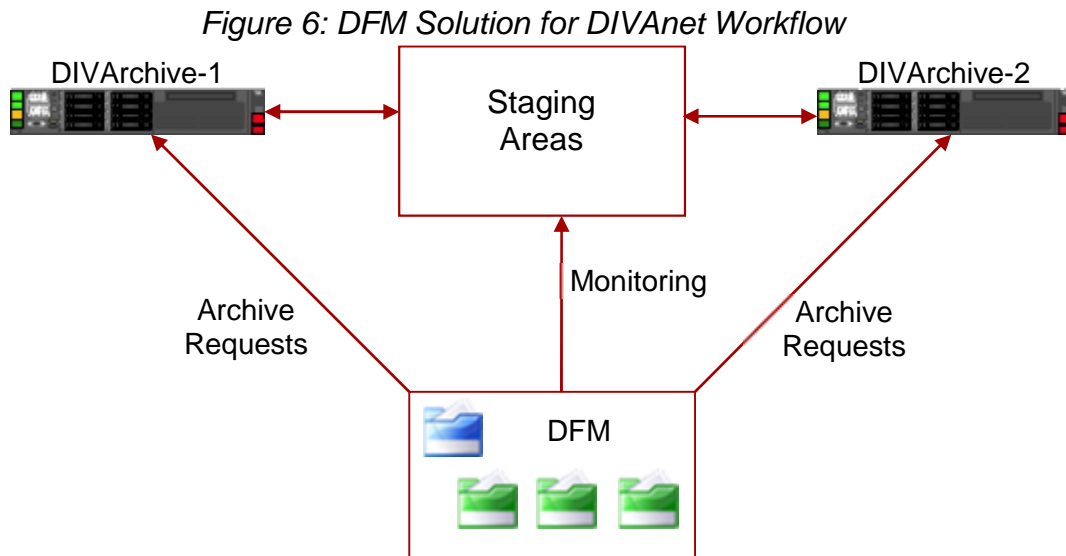
```
DIVA\Program\InterLink\dfm\bin\dfm.bat start
```

2.4 DFM Solution for DIVAnet

This DFM Solution monitors a local disk and is capable of using either Single Files or File Sets. Each Metadata File found in the Drop Folder is parsed and a new object (*usually with more than one file*) is archived in the DIVArchive System.

See A1 *DIVArchive Options and Licensing* for *DIVArchive* licensing information.

This solution is used with DIVAnet Systems that are configured with inter-site transfer capabilities. DIVAnet Contact Oracle Support for detailed information on the various DIVAnet Solutions available.



- Supported Source/Destinations:
 - Local disks.
- Supported objects to be archived:
 - Single and multiple components (*files – non-Complex Objects*).
- Supported major configuration parameters:
 - Manager configuration.
 - Drop Folder URL.
 - DIVArchive Media Name Pattern for generation of Media Name used in archive request.
 - Source/Destination Name
 - Source/Destination Path

2.4.1 Configuration of DFM Solution for DIVAnet

This DFM Solution monitors the local disk. Each Metadata File found in the Drop Folder is parsed and a new object (*usually with more than one file but not Complex Objects*) is archived in DIVArchive System.

See A1 DIVArchive Options and Licensing for DIVArchive licensing information.

Use the following procedure to configure the DFM Solution for DIVAnet.

1. Rename the `dfm.conf.divanet.ini` to `dfm.conf`.
 `DIVA\Program\conf\dfm\dfm.conf`
2. Specify the Manager host and port for `<managerConnetion>`.

Example:

```
<managerConnetion>
  <address host="localhost" port="9000"/>
</managerConnetion>
```

3. Specify the Windows Service Name for `<serviceName>`.
 - This parameter should be written on one line and there should not be any symbols or empty spaces at the beginning of the line.

Example:

```
<serviceName>DIVArchiveDFM</serviceName>
```

4. Specify the `<folderConfig>` node using `<type>`.
 - There are two types of directories: **Single** and **File Set**. If the folder type should be single – the **single** value should be specified, if the folder type should be File Set – the **set** value should be specified as in the example below.

Example:

```
<type>set</type>
```

5. Specify the folder URL using `<url>`.

Example:

```
<url>file:///c:\DROPFOLDER\</url>
```

- Refer to Folder Configuration Examples for folder configuration examples.

6. Specify the DIVArchive Media Name Pattern with `<divaMediaNamePattern>` for generation of the Media Name used in archive requests.

Example:

```
<divaMediaNamePattern>/data/$GROUP/</divaMediaNamePattern>
```

7. Specify the DIVArchive Source/Destination Name.

Example:

```
<sourceDestinationDIVAName>SourceDestinationServer</sourceDestinationDIVAName>
```

8. Specify the DIVArchive Source/Destination Path.

Example:

```
<sourceDestinationDIVAPath>path</sourceDestinationDIVAPath>
```

9. Rename the `DIVA_HOME\Program\conf\dfm\dfm.trace.ini` file to `dfm.trace`.

- **If this step is not completed the installation will fail because it cannot find the trace file.**

10. Install the DFM Service:

```
DIVA\Program\InterLink\dfm\bin\dfm.bat install
```

11. Start the DFM Service:

```
DIVA\Program\InterLink\dfm\bin\dfm.bat start
```

Note: For more information and details about the full solution and configuration, contact Oracle Services to speak with a Service Technician.

3 DFM Key Features and Configuration Options

3.1 DFM Monitoring Workflow and Additional Configuration

3.1.1 Overview and Definitions

DFM constantly monitors the configured Drop Folders. When a file, or File Set, is found in one of the configured Drop Folders it is entered into the internal database (*the internal database is an XML file*) with the file size, date and time of when the file was originally found. After waiting for a specified (*configured*) period of time, DFM will again check the size, time and date of the file (*or File Set*) to see if any changes have occurred.

If the file in a Drop Folder configured in Single File Mode satisfies the applicable criteria in the DFM folder configuration (*i.e.: was not modified in the past 60 seconds*), DFM marks the file as complete and sends an Archive Request to the DIVArchive System requesting that the file be archived. If the file has not fulfilled the criteria for a complete file after a specified (*configured*) allowable time period, then it is marked as incomplete and set for deletion.

In the case of a File Set Folder, once the Metadata (*.mdf*) File is located in the Drop Folder, DFM marks all of the files in the File Set as complete and initiates the archive request to the DIVArchive System. If the Metadata (*.mdf*) File does not exist after a specified (*configured*) period of time then all of the files are marked incomplete and set for deletion.

Caution: The last file to be created in the File Set Drop Folder must be the Metadata File (*.mdf*) which then triggers the archive operation. If the Metadata File is transferred BEFORE the actual data files, content loss can occur!

By default, DFM will try to archive any file(s) that it locates in the configured and monitored Drop Folders. If the files cannot be archived according to the DFM folder configuration, they will be marked as incomplete files. The different reasons for files and File Sets being either archived, marked incomplete or deleted are identified in the following sections.

The time period interval for how often the Drop Folder is scanned is specified in the `fileReloadingThreshold` parameter in the configuration file for that specific folder. Any files or File Sets that are not complete and/or archived within the specified time period will be flagged as an unsuccessful archive, an incomplete object, or deleted.

The DFM internal database is stored as an **XML** file and updated after these different tasks take place. Each time DFM checks the configured Drop Folders, it compares the file date, time, and size to what was previously written to the internal database.

Note: Source files cannot be deleted from a Drop Folder after an archive completes unless the appropriate S/D is configured using the `-allow_delete_on_source` option.

For details on configuration options and features refer to DFM Key Features and Configuration Options, and Folder Configuration Examples.

3.1.2 Algorithm and Workflow for Single File Mode Folders

For **Single File Mode** folders, DFM uses a series of checks to identify if a file is complete and ready for processing, already processed, or incomplete.

To begin the process, DFM first checks to find out if the file size is equal to zero (0) or if the file has been marked as incomplete. If the file size does equal zero, or the file was already marked incomplete, then the file is skipped.

DFM next checks the date of the last modification to the file and the size of the file. If the date of the last modification to the file, or the file size, has changed then the file is skipped. The `lastModificationTime` used in this check is the local DFM time when the last modification of size or date was captured by DFM.

DFM will check to find out if the difference between the current time (*local DFM time*) and the `lastModificationTime` is greater than the `fileReadyThreshold` parameter setting in the configuration file, if the file is locked for processing or not, and if the disk has free space available.

- If the difference is greater than the `fileReadyThreshold` parameter setting, the file is unknown to DFM, not locked for processing, and disk space is available, then the file will become “known” to DFM and DFM tells DIVArchive to archive the file.
- If the difference is greater than the `fileReadyThreshold` parameter setting, the file is already known to DFM, not locked for processing, and should be deleted as incomplete, then DFM will delete the file.
- If the difference is greater than the `fileReadyThreshold` parameter setting, DFM knows about the file for longer than the `incompleteThreshold` parameter setting, and it is not locked for processing, then DFM will mark the file as incomplete.

Note: An “unknown” status indicates that all required information about the file was not added to DFM. DFM stores the date and time information for each file, but if it does not have all of the required information, the file will remain “unknown” to DFM. It will only be “known” once all information about the file has been added to DFM for processing.

Example 1:

1. The file `test.mpg` was found by DFM and the date and size was stored.
2. The file `test.mpg` was found by DFM again (*during the next cycle of folders investigation*), the date and size was changed from the previous time, and therefore the file will not be added as “known” to DFM.
3. The file `test.mpg` was found by DFM again (*during the next cycle of folders investigation*), the date and size was not changed from the previous time, the difference is greater than the `fileReadyThreshold` parameter setting, the file is “unknown” to DFM, not locked for processing, and disk space is available and therefore the file was added as “known” to DFM.
4. The archive request was sent to DIVArchive.
5. The archive was completed.

Example 2:

1. The file `test.mpg` was found by DFM and the date and size was stored.
2. The file `test.mpg` was found by DFM again (*during the next cycle of folders investigation*), the date and size was changed from last time, so the file will not be added as “known” to DFM.
3. The file `test.mpg` was found by DFM again (*during the next cycle of folders investigation*), the date and size was not changed from the previous time, the difference is greater than the `fileReadyThreshold` parameter setting, the file is

unknown to DFM, not locked for processing, and disk space is available so file was added as “known” to DFM.

4. The archive request was sent to DIVArchive.
5. The archive was **not** completed. Therefore, it was aborted by DIVArchive for some reason.
6. DIVArchive tries to archive again after `<repeatRequestDIVAThreshold>` seconds but the request was rejected again.
7. If the difference is greater than the `fileReadyThreshold` parameter setting, DFM knows about the file for longer than the `incompleteThreshold` parameter setting, and it is **not** locked for processing, then DFM will mark the file as incomplete; otherwise the system will continue to retry starting with step 4.

3.1.3 Algorithm and Workflow for File Set Mode Folders

For **File Set Mode** folders DFM uses a series of checks to identify if a file is complete and ready for processing, already processed, or if it is incomplete.

To begin the process, DFM first checks to find out if the file size is equal to zero (0) or if the file has been marked as incomplete. If the file size does equal zero, or the file was already marked incomplete, then the file is skipped.

DFM next checks the date of the last modification to the file and the size of the file. If the date of the last modification to the file, or the file size, has changed then the file is skipped. The `lastModificationTime` used in this check is the local DFM time when the last modification of size or date was captured by DFM.

DFM will check to find out if the difference between the current time (*local DFM time*) and the `lastModificationTime` is greater than the `fileReadyThreshold` parameter setting in the configuration file, if the file is locked for processing or not, and if the disk has free space available.

- If the difference **is** greater than the `fileReadyThreshold` parameter setting, the file is unknown to DFM and not locked for processing, then DFM tells DIVArchive to archive the file.
- If the difference **is** greater than the `fileReadyThreshold` parameter setting, the file is already known to DFM, not locked for processing, the file archived successfully, and the difference between the current time (*local DFM time*) and the archive time is greater than 5 seconds **OR** the file should be deleted as incomplete, then DFM will delete the file.
- If the difference **is** greater than the `fileReadyThreshold` parameter setting, DFM knows about the file for longer than the `incompleteThreshold` parameter setting, and it is not locked for processing, then DFM will mark the file as incomplete.

Note: An “unknown” status indicates that all required information about the file was not added to DFM. DFM stores the date and time information for each file, but if it does not have all of the required information, the file will remain “unknown” to DFM. It will only be “known” once all information about the file has been added to DFM for processing.

Example 1:

1. The file `test.mdf` was found by DFM and the date and size was stored.
2. The file `test.mdf` was found by DFM again (*during the next cycle of folders investigation*), the date and size was changed from the previous time; therefore the file will **not** be added as “known” to DFM.
3. The file `test.mdf` was found by DFM again (*during the next cycle of folders investigation*), the date and size was **not** changed from the previous time, the difference is greater than the `fileReadyThreshold` parameter setting, the file was unknown to DFM and therefore the file was added as “known” to DFM.
4. The archive request is sent to DIVArchive.
5. The archive was completed.
6. The file `test.mdf` was found by DFM again (*during the next cycle of folders investigation*), the date and size was **not** changed from the previous time, the difference is greater than the `fileReadyThreshold` parameter setting, the file is “known” to DFM, not locked for processing, the file archived successfully, and the difference between the current time (*local DFM time*) and the archive time is greater than 5000, therefore it was deleted by DFM.

Example 2:

1. The file `test.mdf` was found by DFM and the date and size was stored.
2. The file `test.mdf` was found by DFM again (*during the next cycle of folders investigation*), the date and size was changed from the previous time, so the file will **not** be added as “known” to DFM.
3. The file `test.mdf` was found by DFM again (*during the next cycle of folders investigation*), the date and size was **not** changed from the previous time, the difference is greater than the `fileReadyThreshold` parameter setting, the file is “unknown” to DFM and therefore the file was added as “known” to DFM.
4. The archive request was sent to DIVArchive.
5. The archive was **not** completed and therefore aborted by DIVArchive for some reason.
6. DIVArchive tries to archive again after `<repeatRequestDIVAThreshold>` seconds but the request was rejected again.
7. If the difference is greater than the `fileReadyThreshold` parameter setting, DFM “knows” about the file for longer than the `incompleteThreshold` parameter setting, and it is **not** locked for processing, then DFM will mark the file as incomplete; otherwise the system will continue to retry starting with step 4.

3.1.4 Incomplete File Threshold (`<incompleteThreshold>`)

If the file is copied into a File Set Mode folder, and during the designated `incompleteThreshold` seconds it was not archived, the file will be marked as **incomplete**.

Example:

```
<incompleteThreshold>86400</incompleteThreshold>
```

3.1.5 Processing of Incomplete Files (<incompleteProcessingStrategy>)

This parameter identifies how DIVArchive will process incomplete files. Incomplete files are files that were placed into the Drop Folder without the Metadata File (*for File Set Folders*) and files that cannot be archived to the DIVArchive System after a specific number of attempts.

The parameters used to identify incomplete files are:

- `incompleteThreshold`
- `maxRejectCountInitializing`
- `maxRejectCountProcessing`

There are 3 options available:

- None
 - This option tells the system not to do anything with an incomplete file.
- Delete
 - This option tells the system to delete the incomplete file.
- Rename
 - This option tells the system to rename the file so it is identifiable as an incomplete archive.

Example:

```
<incompleteProcessingStrategy>OPTION</incompleteProcessingStrategy>
```

Where `OPTION` is one of the three options listed above.

3.1.6 MDF Configuration Priority (<mdfConfigPriority>)

Note: This parameter only applies to File Set folders.

This parameter can have Primary or Secondary values and determines how the system will handle the file during the archive process; either by using the Metadata File parameters (*Primary*) or the folder specific configuration parameters (*Secondary*). The default value used when this parameter is empty or missing is Primary.

- Primary
 - The configuration parameters specified in the Metadata File are used in the archive request.
- Secondary
 - The configuration parameters specified in the folder specific configuration are used in the archive request.

Example:

```
<mdfConfigPriority>Primary</mdfConfigPriority>
```

3.2 Additional Configuration for DIVArchive System Operations

3.2.1 Delete Before Archive (<deleteBeforeArchive>)

Note: This optional parameter may lead to content deletion in the DIVArchive System, so it should be used with caution.

This optional parameter specifies whether or not DFM should delete the object before the archive request for a specific folder.

- This parameter value can be either **TRUE** or **FALSE**.
- The default value is **FALSE**.

Example:

```
<deleteBeforeArchive>FALSE</deleteBeforeArchive>
```

Example:

An object is copied to the Drop Folder and fits into an established Storage Plan which specifies the object should be copied to another S/D and archived to a specific folder. Using this option, the object would be copied to the specified S/D, but then deleted before the Archive Action is performed and therefore never actually archived.

3.2.2 Number of Files to Archive Wildcard (<numberFilesToArchiveWildcard>)

This parameter defines a threshold on the number of files in the same folder. If the number of files in the folder is larger than the **numberFilesArchiveWildcard** parameter, DFM will send the archive request to DIVArchive Manager using the wildcard * as the File List parameter instead of the actual list of files for the entire folder. The value must be an integer to specify the number of files, or if * is used in the request, all files will be archived.

Example:

```
<numberFilesToArchiveWildcard>10</numberFilesToArchiveWildcard>
```

3.2.3 Object Name Conversion Rules (<objectNameConvertRule>)

Note: This parameter only applies to Single File Mode folders

The Object Name Conversion Rules identify how DIVArchive translates filenames into object names.

Currently only the **Simple** conversion method is implemented. The rule syntax for this method is case sensitive and described below.

Note: When you specify a rule using **Name.Ext** it means the Object Name will be exactly the same as the filename. If you specify **Name** (*without the .Ext*) then the filename will be the same as the original filename however the extension will be removed. This applies to all of the parameters listed below.

- **Name.Ext**
 - This is the original filename and extension.
 - No conversion of the filename or extension takes place.

- **name** (all lower case)
 - Convert the original filename to lower case.
 - Remove the file extension.
- **NAME** (*all upper case*)
 - Convert the original filename to all UPPER CASE.
 - Remove the file extension.
- **ext** (*all lower case*)
 - Remove the filename.
 - Convert the original filename to the extension and all lower case.
 - **Example:** `filename.ext` will become simply `.ext`
- **EXT**
 - Remove the filename.
 - Convert the original filename to the extension and all UPPER CASE.
 - **Example:** `filename.EXT` will become simply `.EXT`
- **name.Ext**
 - Convert the original filename to all lower case.
 - Leave the original file extension intact.
- **name.EXT**
 - Convert the original filename to all lower case.
 - Convert the original file extension to all UPPER CASE.
- **pre_Name_suf.EXT**
 - Convert the original filename to `pre_(original filename)_suf.`
 - Convert the original file extension to all UPPER CASE.
- **pre_Name_suf**
 - Convert the original filename to `pre_(original filename)_suf.`
 - Remove the file extension.

Example:

The following uses the **Simple** method and will leave the filename and extension unchanged.

```
<objectNameConvertRule method="simple">Name.Ext</objectNameConvertRule>
```

3.2.4 Original Server (<originalServer>)

In some configurations, the monitored folder is an intermediate location between the video server and the DIVArchive System. In order to make sure that the original content location is set correctly in the DIVArchive System, the `originalserver` parameter can be specified. The original server will be displayed in the GUI and will be stored in the database as the

original content location instead of the actual Source/Destination used to archive data from the intermediate location.

Example:

```
<originalServer>original_server</originalServer>
```

In the example above, `original_server` is the original content location as is designated by the original hostname (e.g. `VIDEO_SERVER_1`).

3.2.5 Original Path (`<originalPath>`)

In some configurations the monitored folder is an intermediate location between the Video Server and the DIVArchive System. In order to make sure that the content path at the original location is set correctly in the DIVArchive System, the `originalPath` parameter can be specified. The original content path will be displayed in the GUI and will be stored in the database as the path to the original content location instead of the actual Root Path used to archive data from the intermediate location.

Example:

```
<originalPath>original_path</originalPath>
```

In the example above, `original_path` is the content path at the `original_server` location.

3.2.6 DIVArchive Media Name (`<divaMediaName>` and `<divaMediaNamePattern>`)

The `<divaMediaNamePattern>` is the DIVArchive Media Name Pattern that will be used to select the medium used in archive requests. Using this pattern allows the `GROUP` to be identified.

Example:

```
<divaMediaNamePattern>folder1/$GROUP/folder3</divaMediaNamePattern>
```

Note: In the example above, `folder1` and `folder3` are actual folder names while `$GROUP` indicates that the system will identify the group from the MDF file.

If the pattern was not recognized, then `<divaMediaName>` from the file set's corresponding MDF file will be used as the DIVArchive Media Name for the archived request when working with a Set Mode Folder. If the pattern is recognized, then the `<divaMediaName>` from the configuration file will be used.

3.3 Additional Configuration for the Monitored Folder

3.3.1 MDF File Extension (`<mdfExtension>`)

This option identifies the extension for the Metadata File for each folder. In the example below, the Metadata File will have the extension `mdf`.

Example:

```
<mdfExtension>mdf</mdfExtension>
```


3.3.2 Source/Destination DIVArchive Path (<sourceDestinationDIVAPath>)

Note: This parameter only applies to File Set Drop Folders. Since the Actor and DFM may be on different machines, they may view the folder differently. For example, Actor may see the folder as a local disk while DFM may see it as an FTP folder or vice versa.

This field defines the location of the **Drop Folder Root** on the DIVArchive Source/Destination. In the case of a dedicated Source/Destination (*the Root Path of the Source/Destination in the Configuration Utility points to the root of DFM*), the **File Path Root** of the Source/Destination may already contain a path, and then this field may be left empty.

Example:

```
<sourceDestinationDIVAPath>path</sourceDestinationDIVAPath>
```

In the example above, `path` would be the full path to the root DIVArchive Source/Destination folder (e.g.: `DROP_FOLDER`).

3.3.3 Archive File Path Template (<archiveFilePathTemplate>)

Note: This parameter is applicable only to Single Drop Folders.

This parameter specifies the rule for how the Root Path of the archive request will be generated.

- The `platform` parameter can be one of the following:
 - WIN
 - SOL
 - CIFS
 - DETECT

The format of the value is a combination of folder names, separators, and the keywords `PARENTn`, `URL_TO_FILE`, and `URL_TO_PARENTn`.

The following keywords are supported:

- `PARENTn`
 - The folder name, where the definition is:
`DFM_FILE_FULL_PATH = DFM_URL \ PARENTn ... \ PARENT 3 \ PARENT 2 \ PARENT 1 \ filename.ext`
- `URL_TO_PARENTm`
 - The relative path from `DFM_URL` to `PARENTm`.
- `URL_TO_FILE`
 - The relative path from `DFM_URL` to the file.

Example:

```
<archiveFilePathTemplate platform="DETECT" options="">  
URL_TO_FILE  
</archiveFilePathTemplate>
```

Note: Refer to Section 5.5 for further details and samples.

3.3.4 Archive Filename Template (<archiveFileNameTemplate>)

Note: This parameter is applicable only to Single Drop Folders.

Using this parameter allows the user to specify the rule to be used for identifying how the filenames of an Archive Request will be generated.

- The `platform` parameter can be one of the following:
 - WIN
 - SOL
 - CIFS
 - DETECT

The format of the value will be combination of folder names, separators, keywords:

- PARENTn
- URL_TO_FILE
- Keywords

The following keywords are supported:

- PARENTn
 - The folder name, where the definition is:
`DFM_FILE_FULL_PATH = DFM_URL \ PARENTn ... \ PARENT3 \ PARENT2 \ PARENT1 \ filename.ext`
- URL_TO_FILE
 - The relative path from `DFM_URL` to the file.
- `filename.ext`
 - The original filename.
- `filename`
 - The filename without an extension.

Example:

```
<archiveFileNameTemplate platform="DETECT" options="">
  filename.ext
</archiveFileNameTemplate>
```

Note: In the example above `filename` and `ext` represent the keywords of the pattern definition, not the sample filename and extension. `filename.ext` is not an actual filename. Refer to 5.5 for further details and samples.

3.3.5 Source/Destination Path Samples for Different Configurations

In File Set mode, the `sourceDestinationDivaPath` will be added to the relative file path.

Examples:

If the `sourceDestinationDivaPath = E:\DFM:`

Archive File Path Root = E:\DFM\data\tape1\123456

Archive File List = toto.mdf toto0.data folder1\toto1.data
folder1\folder2\toto2.data

The results will be as follows:

- E:\DFM\data\tape1\123456\toto.mdf
- E:\DFM\data\tape1\123456\toto0.data
- E:\DFM\data\tape1\123456\folder1\toto1.data
- E:\DFM\data\tape1\123456\folder1\folder2\toto2.data

If the SourceDestinationDivaPath = F:\DFM\:

Archive File Path Root = F:\DFM\data\tape1\123456

Archive File List = toto.mdf toto0.data folder1\toto1.data
folder1\folder2\toto2.data

The results will be as follows:

- F:\DFM\data\tape1\123456\toto.mdf
- F:\DFM\data\tape1\123456\toto0.data
- F:\DFM\data\tape1\123456\folder1\toto1.data
- F:\DFM\data\tape1\123456\folder1\folder2\toto2.data

If the SourceDestinationDivaPath is empty:

Archive File Path Root = data\tape1\123456\

Archive File List = toto.mdf toto0.data folder1\toto1.data
folder1\folder2\toto2.data

The results will be as follows:

- data\tape1\123456\toto.mdf
- data\tape1\123456\toto0.data
- data\tape1\123456\folder1\toto1.data
- data\tape1\123456\folder1\folder2\toto2.data

If the SourceDestinationDivaPath = data:

Archive File Path Root = data\data\tape1\123456\

Archive File List = toto.mdf toto0.data folder1\toto1.data
folder1\folder2\toto2.data

The results will be as follows:

- data\data\tape1\123456\toto.mdf

- data\data\tape1\123456\toto0.data
- data\data\tape1\123456\folder1\toto1.data
- data\data\tape1\123456\folder1\folder2\toto2.data

Refer to Section 5.5 for examples of configuration files for **Single Mode Folders**.

3.3.6 File Filter (<fileFilter>)

This parameter is used to filter files to archive. The `type` attribute can be specified as either `include` or `exclude`.

In **inclusion mode**, no files will be archived if there are no filters defined. Each filter will allow archiving of files if they match specific rules. The file will only be archived if the file corresponds to one or more masks. If the file does not correspond to any masks then the file will be ignored by DFM and not deleted nor archived.

In **exclusion mode**, all files will be archived if there are no filters defined. Each filter will restrict the archiving of files if they match specific rules. The file will **not** be archived if the file corresponds to one or more masks. If the file corresponds to any mask, the file will be ignored by DFM and not deleted nor archived.

The filter can be defined as a pattern where the mask has a question mark (?) indicating any symbol or an asterick (*) indicating any string, as a Regular Expression (*RegExp*), or both.

The following parameters are used within the `<fileFilter>` configuration parameter:

- `<mask>`
 - Specifies the pattern for the filter.
- `<regexp>`
 - Specifies the regular expression for the filter.

```
<fileFilter type="exclude">
<mask>a*.mov</mask>
<regexp>.[hc]at</regexp>
</fileFilter>
```

Note: The `<fileFilter>` parameter is optional for the Drop Folder configuration. If it is not specified in the folder configuration, DFM will archive any file in the folder without using any masks. Only one `<fileFilter>` can be specified per folder. For details on Regular Expressions (*RegExp*), visit:

- http://en.wikipedia.org/wiki/Regular_expression
- <http://www.regular-expressions.info>

3.3.7 Delete Parent Directory and Content Directories (<deleteParentDirectoryAndContentDirectories>)

This optional parameter specifies whether DFM should delete the parent and child folders of an MDF file. This is useful in the case where a user puts an object into a folder and DFM is NOT set to delete the files/folders after all actions are executed on the object. In this manner, the object's files and folders are not left in the folder where the object was placed.

- This parameter can have a value of **TRUE** or **FALSE**.
- The default value is **TRUE**.

Example:

Directories will NOT be deleted:

```
<deleteParentDirectoryAndContentDirectories>  
FALSE  
</deleteParentDirectoryAndContentDirectories>
```

Directories WILL be deleted

```
<deleteParentDirectoryAndContentDirectories>  
TRUE  
</deleteParentDirectoryAndContentDirectories>
```

4 Folder Configuration Examples

Note: The examples in the following sections have had all comments that are in the actual configuration files removed to make them easier to understand.

4.1 Example of a Local Folder in File Set Mode

```
<folderConfig>
  <url>file:///c:\DROPFOLDER</url>
  <type>set</type>
  <priority>50</priority>
  <categoryName>category1</categoryName>
  <incompleteThreshold>60</incompleteThreshold>
  <sourceDestinationDIVAName>LOCAL1</sourceDestinationDIVAName>
  <sourceDestinationDIVAPath>C:/DROPFOLDER</sourceDestinationDIVAPath>
  <divaMediaName>Array1</divaMediaName>
  <incompleteProcessingStrategy>Delete</incompleteProcessingStrategy>
</folderConfig>
```

In the example configuration above:

- The local folder is **c:\DROPFOLDER**.
- The type of DFM folder is **File Set**.
- The Priority is **50**.
- The Category is **category1**.
- The Incomplete Threshold is **60** seconds.
- The Source/Destination DIVArchive Name is **LOCAL1**.
- The Source/Destination DIVArchive Path is **C:/DROPFOLDER**.
- The DIVArchive Media Name is **Array1**.
- The Incomplete File Processing Strategy is **Delete**.
 - All incomplete folders will be deleted.

Note: the Object Name Conversion Rule does not apply to Set Mode Folders. Files will be acted upon using the original filename and extension.

Example:

Using the configuration above, if a set of files is copied into the DFM-monitored local folder named **c:\DROPFOLDER**, and has 3 files in the set with the names:

- **file1.ext**
- **file2.ext**
- **filesetdata.mdf**

They will be archived to a Single Object in the DIVArchive System with the following parameters:

- Source/Destination: LOCAL1
- Object Name: `filesetdata.ext`
- Category Name: `category1`

If the Metadata File is not there, the set will be considered incomplete and the files will all be deleted after 60 seconds.

4.2 Example of an FTP Folder in Single File Mode

When the DFM Drop Folder is configured on an FTP Server, all content (*files and folders expected to be archived*) should be *copied to DFM Drop Folder via FTP*. If the content is copied to the DFM Drop Folder by other means (*local copy via OS, remote desktop copy, etc.*) there is no guarantee of proper processing by DFM.

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

User: diva

User Home Directory: `/ifs`

Folder to be Monitored: `/ifs/folder1`

Correct DFM Configuration: `ftp://diva:password@host_ip/folder1`

Incorrect DFM Configuration: `ftp://diva:password@host_ip/ifs/folder1`

```
<folderConfig>
  <url>ftp://diva:diva@localhost:21/dropfolder</url>
  <type>single</type>
  <priority>20</priority>
  <categoryName>category2</categoryName>
  <incompleteThreshold>120</incompleteThreshold>
  <sourceDestinationDIVAName>FTP1</sourceDestinationDIVAName>
  <sourceDestinationDIVAPath>/video1/data/</sourceDestinationDIVAPath>
  <divaMediaName>Array1</divaMediaName>
  <objectNameConvertRule method="simple">Name</objectNameConvertRule>
  <divaMediaNamePattern>/def/$GROUP/content</divaMediaNamePattern>
</folderConfig>
```

In the example above:

- The FTP folder is `ftp://diva:diva@localhost:21/dropfolder`.
- The Type of DFM folder is Single.
- The Priority is 20.
- The Category is category2.

- The Incomplete Threshold is 120 seconds.
- The Source/Destination DIVArchive Name is FTP1.
- The Source/Destination DIVArchive Path is /video1/data/.
- The DIVArchive Media Name is Array1.
- The default Object Name conversion rule for Drop Folders will remove the file extension.
- The `divaMediaNamePattern` is `/def/$GROUP/content`.

Example:

Using the configuration above, if a single file is transferred to the FTP server called `localhost` on port 21, with the login name `diva` and the password `diva`, and in the DFM-monitored FTP folder named `dropfolder` with the filename `file1.ext`.

It will be archived to a Single Object in the DIVArchive System with the following parameters:

- Source/Destination: `FTP1`
- Object Name: `file1`
- Category Name: `category2`

If the file is not completely present, or if the archive process does not complete, the file will be considered incomplete and deleted after 120 seconds.

4.3 Example of a CIFS Folder

```
<folderConfig>
  <url>file:///\\divaserver\dropfolder</url>
  <type>set</type>
  <priority>50</priority>
  <categoryName>category3</categoryName>
  <incompleteThreshold>60</incompleteThreshold>
  <sourceDestinationDIVArchiveName>CIFS1</sourceDestinationDIVArchiveName>
  <sourceDestinationDIVArchivePath></sourceDestinationDIVArchivePath>
  <divaMediaName>array1</divaMediaName>
  <incompleteProcessingStrategy>Rename</incompleteProcessingStrategy>
</folderConfig>
```

In the example above:

- The CIFS folder is `\\divaserver\dropfolder`.
- The Type of DFM folder is **File Set**.
- The Priority is **50**.
- The Category is **category3**.

- The Incomplete Threshold is **60** seconds.
- The Source/Destination DIVArchive Name is **CIFS1**.
- The Source/Destination DIVArchive Path is **empty**.
- The DIVArchive Media Name is **array1**.
- The incomplete file processing strategy is **Rename**.
 - All incomplete folders will be renamed to **failed_ + (original filename)**.

Example:

Using the configuration above, if a set of files is copied into the DFM monitored CIFS folder named `dropfolder` on the machine called `divaserver`, and has 3 files in the set with the names:

- `file1.ext`
- `file2.ext`
- `filesetdata.mdf`

They will be archived to a Single Object in the DIVArchive System with the following parameters:

- Source/Destination: `CIFS1`
- Object Name: `filesetdata.ext`
- Category Name: `category3`

If the Metadata File is not there, the set will be considered incomplete and the files will all be deleted after 60 seconds.

Note: the Object Name Conversion Rule does not apply to Set Mode Folders. Files will be acted upon using the original filename and extension.

4.4 Tracing Configuration

To specify the Tracing Configuration use the following parameters:

- `wrapper.logfile.loglevel`
 - Specifies the level of messages to be logged.
- `wrapper.logfile.maxsize`
 - Specifies the maximum size (*in bytes, kilobytes, or megabytes*) of the log file.
- `wrapper.logfile.maxfiles`
 - Specifies the maximum log files count.

The following is a sample log file configuration:

```
# Log Level for log file output. (See docs for log levels)
wrapper.logfile.loglevel=INFO

# Maximum size that the log file will be allowed to grow to before
```

```
# the log is rolled. Size is specified in bytes. The default value  
# of 0 disables log rolling. May abbreviate with the 'k' (kb) or  
# 'm' (mb) suffix. For example: 10m = 10 megabytes.  
wrapper.logfile.maxsize=10m
```

```
# Maximum number of rolled log files which will be allowed before  
# old files are deleted. The default value of 0 implies no limit.  
wrapper.logfile.maxfiles=4
```

Valid logging levels are:

- **NONE:** No output.
- **FATAL:** Show only fatal error messages.
- **ERROR:** Show all error messages.
- **WARN:** Show all warning messages.
- **STATUS:** Show all state changes.
- **INFO:** Show all JVM output and informative messages.
- **DEBUG:** Show detailed debug information.

5 Advanced DFM Configuration

Note: These parameters should not be altered unless required and the operator is confident of having the ability to do so correctly. Misconfiguration of these parameters could result in an unstable and/or non-functioning system! It is recommended that the original configuration file be saved to a different filename before making any modifications to the configuration (e.g. *dfm.conf.original.ini*).

5.1 DFM Full Configuration File

There are several groups of parameters for the configuration file:

- Global DFM Parameters
 - Used to specify common behaviour of the DFM.
- Folders Default Parameters
 - Used to specify some default values for folder-specific parameters.
- Folder-Specific Parameters
 - Used to specify folder-specific parameters.
- Tracing Specific Parameters
 - Used to specify tracing parameters.

Refer to Appendix A7 Sample of Full DFM Configuration File (*dfm.conf.ini*) to view a sample of the full configuration file.

5.2 Global DFM Configuration

The DFM configuration XML provides the following elements for connections configuration:

Table 3: Global DFM Configuration Parameters

Parameter	Definition
<code><managerConnetion></code>	<p>Specifies how DFM will connect to Manager. It is not required that these options are defined unless the DIVArchive System being connected uses connection parameters outside of the default configuration. See <i>A1 DIVArchive Options and Licensing for DIVArchive licensing information</i>.</p> <p>Example:</p> <pre><address host="..." port="..." userName="..." password="..." applicationName="..." /></pre> <p>host (<i>required</i>): The address of the DIVArchive Manager.</p> <p>port (<i>required</i>): The port on which the DIVArchive Manager is listening. This should be a positive integer value.</p> <p>userName (<i>optional</i>): The user name that will be used to log into the DIVArchive System. If <code>NULL</code> or empty, DIVArchive will use the default user name and privileges (<i>from the DIVArchive System</i>) for this application.</p> <p>password (<i>optional</i>): The password for the DIVArchive user. The connection will be rejected if the password is incorrect.</p> <p>applicationName (<i>optional</i>): An arbitrary string that is interpreted by DIVArchive as an application name. This is used for information purposes only and may be <code>NULL</code> or an empty string.</p>
<code><serviceName></code>	<p>Service Name by which the current instance of DFM will be installed. Spaces are not allowed within the Service Name.</p> <p>The Service Name will be:</p> <p style="text-align: center;"><code>DIVArchive DFM - serviceName</code></p> <p>If <code>serviceName</code> is left empty, the Service Name will be <code>DIVArchive DFM</code>.</p>
<code><foibStateFile></code>	Filename of the File/Object Information Base dump.
<code><doibStateFile></code>	Filename of the Delete File/Object Information Base dump.

Parameter	Definition
<code><fileReloadingThreshold></code>	File reloading threshold in seconds. This interval specifies how often the File Monitor Module will reload the state of the DFM folders. This setting must be a positive integer value between 1 and 10,000,000.
<code><fileReadyThreshold></code>	Time interval, in seconds. If a file's last modification time to the file remains unchanged after this interval the file will be marked as "copied" and then archived. This setting must be a positive integer value between 1 and 10,000,000.
<code><dirDeleteThreshold></code>	Directory Delete Threshold in seconds. This interval specifies how often the File Monitor Module will reload the state of the DFM folders in Delete Mode and commence with deletion. This setting must be a positive integer value between 1 and 10,000,000.
<code><fileDeleteThreshold></code>	File Delete Threshold in seconds. This interval specifies when a file will be deleted after copying is complete for DFM folders in Delete Mode. This setting must be a positive integer value between 1 and 10,000,000.
<code><maxArchiveOperations></code>	Maximum number of concurrent archive operations that can be performed by DFM simultaneously. If the number of requests exceeds this threshold, pending requests must wait until all active archive operations are completed. This setting must be a positive integer value between 1 and 10,000,000.
<code><maxRejectCountInitializing></code>	Maximum number of archive retries that can be performed by DFM for requests rejected by DIVArchive during request initialization. Objects exceeding this limit will be marked as incomplete. This setting must be a positive integer value between 1 and 10,000,000.
<code><maxRejectCountProcessing></code>	Maximum number of archive retries that can be performed by DFM for requests rejected by DIVArchive during request processing. Objects exceeding this limit will be marked as incomplete. This setting must be a positive integer value between 1 and 10,000,000.
<code><pollingIntervalOfRequestState></code>	This parameter specifies the frequency of object Status polling in seconds. This setting must be a positive integer value between 1 and 10,000,000.
<code><repeatRequestDIVAThreshold></code>	Interval, in seconds, before DIVArchive's Archive Module will re-submit rejected requests. This setting must be a positive integer value between 1 and 10,000,000.

5.3 Folders Default Configuration

The Folders Default Configuration parameter value is used for folders in a folder-specific configuration when the parameter is not specified. All default values can be overridden within the folder-specific configuration.

The Folders Default configuration accepts the following parameters:

Table 4: Folders Default Configuration Parameters

Parameter	Definition
<code><type></code>	<p>There are two types of directories: Single File and File Set.</p> <p>To configure the folder for Single Files, specify single.</p> <p>To configure the folder for File Sets, specify set.</p>
<code><mode></code>	<p>DFM folder work mode. This can be Archive or Delete.</p>
<code><priority></code>	<p>Default priority of the requests for the folder. The folder priority range is [0...100].</p> <p>The value 0 is the lowest priority and 100 the highest.</p>
<code><categoryName></code>	<p>Default value for Category Name. This value will be provided to DIVArchive if it is not overridden in the folder-specific configuration or in the File Set Metadata File.</p>
<code><incompleteThreshold></code>	<p>Incomplete Threshold in seconds. This parameter specifies the maximum amount of time that a Single File/File Set can reside in the Drop Folder and not be archived before it is marked as incomplete. This should be a positive integer value between 1 and 10,000,000.</p>
<code><sourceDestinationDIVAName></code>	<p>The DIVArchive Source/Destination Name to use in the archive request.</p>
<code><sourceDestinationDIVAPath></code>	<p>The DIVArchive Source/Destination Path to use in the archive request.</p>
<code><mdfExtension></code>	<p>Option to configure the extension for the Metadata File for each folder.</p>

Parameter	Definition
<code><originalServer></code>	DIVArchive Source/Destination to store in the object catalog if different from the actual Source/Destination. This is useful when an intermediate storage area was used for archiving but the user wants to store the original source for the data in the object catalog information.
<code><originalPath></code>	DIVArchive File Root Path to store in the object catalog, if different from the actual File Root Path. This is useful when an intermediate storage area was used for archiving but the user wants to store the original source for the data in the object catalog information.
<code><incompleteProcessingStrategy></code>	Incomplete folder/file processing strategy. The value could be None , Delete , or Rename .
<code><numberOfFilesToArchiveWildcard></code>	Defines the threshold on the number of files in the same folder. If the number of files in the folder becomes larger than the <code>numberOfFilesToArchiveWildcard</code> value, DFM will send the archive request to the Manager using the wildcard (*) for the entire folder. The value must be a positive integer.
<code><divaMediaNamePattern></code>	The DIVArchive Media Name Pattern allows specifying the archive medium by extracting a part of the archive Root Path. This option is mutually exclusive with the <code>divaMediaName</code> option.
<code><deleteBeforeArchive></code>	<p>This parameter specifies whether DFM should delete the object from DIVArchive before the archive operation if the object already exists. This parameter can be set to either TRUE or FALSE.</p> <p>This is an optional parameter and the default value is FALSE.</p>

Parameter	Definition
<pre> <objectNameConvertRule method="simple"> Name.Ext </objectNameConvertRule> </pre>	<p>The value of <code><objectNameConvertRule></code> will specify the rule itself for the algorithm specified.</p> <p>In the current version of DFM, only the Simple conversion method is implemented. The rule syntax for this method is described below:</p> <ul style="list-style-type: none"> • <code>Name.Ext</code> <ul style="list-style-type: none"> ○ Original filename and extension. • <code>name</code> <ul style="list-style-type: none"> ○ <code>toLowerCase(name)</code> • <code>NAME</code> <ul style="list-style-type: none"> ○ <code>toUpperCase(name)</code> • <code>ext</code> <ul style="list-style-type: none"> ○ <code>toLowerCase(extension)</code> • <code>EXT</code> <ul style="list-style-type: none"> ○ <code>toUpperCase(extension)</code> • <code>name.Ext</code> <ul style="list-style-type: none"> ○ <code>toLowerCase(original filename).(original extension)</code> • <code>name.EXT</code> <ul style="list-style-type: none"> ○ <code>toLowerCase(original filename).toUpperCase(original extension)</code> • <code>pre_Name_suf.EXT</code> <ul style="list-style-type: none"> ○ <code>pre_(original filename)_suf.toUpperCase(original extension)</code> • <code>pre_Name_suf</code> <ul style="list-style-type: none"> ○ <code>pre_(original filename)_suf</code>

5.4 Folder-Specific Configuration

All folders monitored by DFM must be described in the configuration file. Folder-specific configuration accepts the following parameters:

Table 5: Folder Specific Configuration Parameters

Parameter	Definition
<code><url></code>	DFM folder URL. This is the monitored folder as viewed by the DFM process.
<code><type></code>	There are two types of directories: Single File and File Set . To configure the folder for Single Files , specify single . To configure the folder for File Sets , specify set .
<code><mode></code>	DFM folder work mode. This can be Archive or Delete .
<code><priority></code>	Priority of the requests for this folder. The folder priority range is [0...100]. The value 0 is the lowest priority and 100 the highest.
<code><categoryName></code>	Value for Category Name. This value is provided to DIVArchive if it is not overridden in a File Set Metadata File.
<code><incompleteThreshold></code>	Incomplete Threshold, in seconds. This parameter specifies the maximum amount of time that a Single File/File Set can reside in the Drop Folder (<i>and not be archived</i>) before it is marked as incomplete. It should be a positive integer value between 1 and 10,000,000.
<code><sourceDestinationDIVName></code>	DIVArchive Source/Destination Name to use in the archive request.
<code><sourceDestinationDIVPath></code>	DIVArchive Source/Destination Path to use in the archive request.
<code><divaMediaName></code>	DIVArchive Media Name to use in the archive request.
<code><mdfExtension></code>	Option to configure the extension for the Metadata File for each folder.
<code><originalServer></code>	DIVArchive Source/Destination to store in the object catalog, if different from the actual Source/Destination. This is useful when an intermediate storage area was used for archiving but the user wants to store the original source of the data in the object catalog information.

Parameter	Definition
<code><originalPath></code>	DIVArchive File Root Path to store in the object catalog, if different from the actual File Root Path. This is useful when an intermediate storage was used for archiving but the user wants to store the original source of the data in the object catalog information.
<code><incompleteProcessingStrategy></code>	Incomplete folder/file processing strategy. This value could be None , Delete , or Rename .
<code><numberFilesToArchiveWildcard></code>	Defines the threshold of the number of files allowed in the same folder for archiving. If the number of files in the folder becomes larger than the <code>numberFilesArchiveWildcard</code> , DFM will send the archive request to the Manager using the wildcard (*) for the entire folder. This value must be an integer.
<code><divaMediaNamePattern></code>	The DIVArchive Media Name Pattern allows specifying the archive medium by extracting a part of the archive Root Path. This option is mutually exclusive with the <code>divaMediaName</code> option.
<code><deleteBeforeArchive></code>	This parameter specifies whether or not DFM should delete the object from DIVArchive before the archive operation if the object already exists. This parameter can be set to either TRUE or FALSE . This is an optional parameter. The default value is FALSE .

Parameter	Definition
<pre> <objectNameConvertRule method="simple">Name.Ext </objectNameConvertRule> </pre>	<p>The value of <code><objectNameConvertRule></code> will specify the rule itself for the algorithm specified.</p> <p>In the current version of DFM, only the Simple conversion method is implemented. The rule syntax for this method is described below:</p> <ul style="list-style-type: none"> • <code>Name.Ext</code> <ul style="list-style-type: none"> ○ Original filename and extension. • <code>name</code> <ul style="list-style-type: none"> ○ <code>toLowerCase(name)</code> • <code>NAME</code> <ul style="list-style-type: none"> ○ <code>toUpperCase(name)</code> • <code>ext</code> <ul style="list-style-type: none"> ○ <code>toLowerCase(extension)</code> • <code>EXT</code> <ul style="list-style-type: none"> ○ <code>toUpperCase(extension)</code> • <code>name.Ext</code> <ul style="list-style-type: none"> ○ <code>toLowerCase(original filename).(original extension)</code> • <code>name.EXT</code> <ul style="list-style-type: none"> ○ <code>toLowerCase(original filename).toUpperCase(original extension)</code> • <code>pre_Name_suf.EXT</code> <ul style="list-style-type: none"> ○ <code>pre_(original filename)_suf.toUpperCase(original extension)</code> • <code>pre_Name_suf</code> <ul style="list-style-type: none"> ○ <code>pre_(original filename)_suf</code> <p>Note: When a rule such as <code>Name.Ext</code> has been specified, it means no changes to the original object name. If <code>Name</code> has been specified, it means to remove the extension and so on.</p>

5.5 Advanced Configuration of File Root Path in Single File Mode

The DFM configuration receives file information from the remote or local media (*FTP Server, Shared Directory, Mounted Drive, etc.*). Based on the filename and folder, DFM generates the DIVArchive File Root Path and DIVArchive Filename.

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

Because requirements for the DIVArchive filename vary, the system can either **include** or **exclude** the parent folder in the file.

The requirements for the path also vary because there are a number of ways the DIVArchive Actor can connect to the media.

The **File Root Path** in **Single File** mode should be specified in the following format:

```
<archiveFilePathTemplate platform="DETECT"
options="">URL_TO_FILE</archiveFilePathTemplate>
```

The filename in **Single File** mode should be specified in the following format:

```
<archiveFileNameTemplate platform="DETECT"
options="">filename.ext</archiveFileNameTemplate>
```

The DFM monitors a DFM URL (*FTP, CIFS, Windows Disk, or Solaris Disk*). For each incoming file DFM will obtain a File Full Path (*for FTP, CIFS, Windows Disk, or Solaris Disk*).

```
File Full Path = DFM_URL \ PARENTn ... \ PARENT3 \ PARENT2 \ PARENT1 \
Filename.Ext
```

or

```
File Full Path = URL_TO_PARENTm \ PARENTm ... \ PARENT3 \ PARENT2 \
PARENT1 \ Filename.Ext
```

Note: Whether “/” or “\” is used will depend on the operating system

```
<archiveFilePathTemplate platform="DETECT"
options="">x:\ccc\ddd</archiveFilePathTemplate>
```

The format of the value is a combination of folder names, separators, and keywords for PARENTn, URL_TO_FILE, and URL_TO_PARENTn.

```
<archiveFileNameTemplate platform="DETECT"
options="">filename.ext</archiveFileNameTemplate>
```

The format of the value is a combination of folder names, separators, and keywords for PARENTn, URL_TO_FILE, and Filename.Ext.

Where the platform is WIN | SOL | CIFS | DETECT.

Available keywords are:

- PARENTn
 - The folder name where the definition is located.
 - File Full Path = DFM URL \ PARENTn ... \ PARENT 3 \ PARENT 2 \ PARENT 1 \ Filename.Ext
- URL_TO_PARENTm
 - The path from the DFM URL to PARENTm.

- `URL_TO_FILE`
 - The path from the DFM URL to the file.
- `Filename.Ext`
 - Original filename.
- `Filename`
 - Filename without an extension.

Note: In all of the following examples it is important to use the correct case because the filename and extension parameters are case sensitive. For example, `filename.ext` is not the same as `Filename.Ext`.

Examples:

Workflow A: The files are located in the root of the DFM URL.

- Drop Folder is on CIFS.
- Actor is on Windows.
 - DFM URL** = `file:///\\aaa\\bbb\\ccc\\ddd`
 - File Full Path** = `\\aaa\\bbb\\ccc\\ddd\\ffffff.txt`
- Actor sees the **DFM URL** as `x:\\ccc\\ddd`.
- The configuration will be as follows:
 - `ARCHIVE_FILE_PATH_TEMPLATE` = `"x:\\ccc\\ddd"`
 - `ARCHIVE_FILE_NAME_TEMPLATE` = `"Filename.Ext"`
- `Filename.Ext` must support both files without the extension and specifies keeping the extension if there is one.

In the case where some files are located in the root of the `DFM_URL` and some are located in sub-folders, the path length is not constant.

This case can be addressed by using:

- `ARCHIVE_FILE_NAME_TEMPLATE` = `"URL_TO_FILE\\Filename.Ext"`
 - When the `URL_TO_FILE` parameter is empty, DFM automatically removes the `"\"` character.
 - Empty `ARCHIVE_FILE_PATH_TEMPLATE` is supported.

Workflow B: The files are located one folder deeper than the root of the DFM URL and the filenames in the File List for the archive request should **NOT** contain the folder names.

- Drop Folder is on FTP.
 - **When the DFM Drop Folder is configured on an FTP Server, all content (*files and folders expected to be archived*) should be copied to DFM Drop Folder via FTP. If the content is copied to the DFM Drop Folder by other means (*local copy via OS, remote desktop copy, etc.*) there is no guarantee of proper processing by DFM.**

- Actor is on UNIX.

DFM URL = ftp://user:password@aaa/bbb/ccc/ddd

File Full Path = /aaa/bbb/ccc/ddd/eee/ffffffff.txt

- Actor sees the **DFM URL** as /ccc/ddd.

- The configuration will be as follows:

ARCHIVE_FILE_PATH_TEMPLATE = "/ccc/ddd/PARENT1"

ARCHIVE_FILE_NAME_TEMPLATE= "Filename.Ext"

- This case must be addressed by using:

ARCHIVE_FILE_PATH_TEMPLATE = "/ccc/ddd/URL_TO_FILE"

Workflow C: The files are located one folder deeper than the root folder of the DFM URL and the filenames in the File List for the archive request should contain the folder names.

- Drop Folder is on FTP.
 - **When the DFM Drop Folder is configured on an FTP Server, all content (*files and folders expected to be archived*) should be copied to DFM Drop Folder via FTP. If the content is copied to the DFM Drop Folder by other means (*local copy via OS, remote desktop copy, etc.*) there is no guarantee of proper processing by DFM.**

- Actor is on UNIX.

DFM URL = ftp://user:password@aaa/bbb/ccc/ddd

File Full Path = /aaa/bbb/ccc/ddd/eee/ffffffff.txt

- Actor sees the **DFM URL** as /ccc/ddd.

- The configuration will be as follows:

ARCHIVE_FILE_PATH_TEMPLATE = "/ccc/ddd"

ARCHIVE_FILE_NAME_TEMPLATE= "PARENT1/Filename.Ext"

- This case must be addressed by using:

ARCHIVE_FILE_NAME_TEMPLATE= "URL_TO_FILE/Filename.Ext"

Workflow D: The files are located in any number of directories deeper than the root of the DFM URL should not contain the folder names.

- Drop Folder is on Windows.

- Actor is on UNIX.

DFM URL = file:///c:\aaa\bbb\ccc\ddd\

File Full Path = c:\aaa\bbb\ccc\ddd\p1\p2\p3\p4\p5\ffffffff.txt

- Actor sees the **DFM URL** as root /.

- The configuration will be as follows:

ARCHIVE_FILE_PATH_TEMPLATE = "/URL_TO_FILE/"

ARCHIVE_FILE_NAME_TEMPLATE= "Filename.Ext"

Workflow E: The files are located any number of directories deeper than the root of the **DFM URL** and the filenames in the File List for the archive request should contain 2 folder names.

- Drop Folder is on Windows.
- Actor is on UNIX.

DFM URL = `file:///c:\aaa\bbb\ccc\ddd\`

File Full Path = `c:\aaa\bbb\ccc\ddd\p1\p2\p3\p4\p5\ffffffff.txt`

- Actor sees the **DFM URL** as root /.
- The configuration will be as follows:

`ARCHIVE_FILE_PATH_TEMPLATE = "/URL_TO_PARENT2/"`

`ARCHIVE_FILE_NAME_TEMPLATE= "PARENT2/PARENT1/Filename.Ext"`

Workflow E-2: Using the example above with the assumption that Actor points to the folder named "ddd".

DFM URL = `file:///c:\aaa\bbb\ccc\ddd\`

File Full Path = `c:\aaa\bbb\ccc\ddd\p1\p2\p3\p4\p5\ffffffff.txt`

- Actor sees the **DFM URL** as root /ddd/.
- The configuration will be as follows:

`ARCHIVE_FILE_PATH_TEMPLATE = "/ddd/URL_TO_PARENT2/"`

`ARCHIVE_FILE_NAME_TEMPLATE= "PARENT2/PARENT1/Filename.Ext"`

Workflow F: If DFM monitors the video server using FTP, the file should have a file extension, and Actor will archive via the server interface without a file extension.

- The file should be located in one folder level.
- Drop Folder is on FTP.
 - **When the DFM Drop Folder is configured on an FTP Server, all content (*files and folders expected to be archived*) should be copied to DFM Drop Folder via FTP. If the content is copied to the DFM Drop Folder by other means (*local copy via OS, remote desktop copy, etc.*) there is no guarantee of proper processing by DFM.**

- Actor using special interface.

DFM URL = `ftp://user:password@aaa/bbb/ccc/ddd`

File Full Path = `/aaa/bbb/ccc/ddd/eee/ffffffff.txt`

- Actor sees the **DFM URL** as "special_#*/ddd".
- The configuration will be as follows:

`ARCHIVE_FILE_PATH_TEMPLATE = "special_#*/ddd/URL_TO_PARENT1"`

`ARCHIVE_FILE_NAME_TEMPLATE= "PARENT1/filename"`

- This is how the file extension is removed.

6 DFM Operation, Administration and Monitoring

6.1 Overview of Starting, Stopping and Restarting DFM

The DFM Service can be started, stopped, and restarted using the Operating System Services facilities or the DFM command line utility:

```
dfm {start|stop|restart}
```

Where:

`start` tells the DFM Service to start.

`stop` tells the DFM Service to stop.

`restart` tells the DFM Service to restart.

6.1.1 Starting and Restarting DFM

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 has performed 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.

Once all of these have completed, the DFM status will be **Running**.

Note: For successful starting of DFM, the `.ini` extension must be removed from the `dfm.trace.ini` file.

6.1.2 Running DFM

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

DFM logs information about the incomplete files and calls the DFM File Manager Module to move them to the Trash Folder.

If DIVArchive requests fail, the DIVArchive Status Module informs the internal database about the failure. If the number of unsuccessful request attempts reaches a pre-configured number, the object status is changed to **could not be archived** and the object is marked as incomplete.

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

WARNING: The last file to be created in the File Set Drop Folder must be the Metadata File (.mdf) which then triggers the archive operation. If the Metadata File is transferred BEFORE the actual data files, content loss can occur!

6.1.3 Stopping DFM

DFM will shut down upon execution of the shutdown script, stopping all internal processes before all archive operations are completed. After all of the modules are stopped, all internal statuses are saved on the disk in the internal database before the DFM completes shutdown.

6.2 Summary of DFM Administration

- Service Installation
 - Installed with DIVArchive in the same installation package.
 - `DIVA_HOME\Program\InterLink\dfm\bin\dfm.bat install`
- Licensing
 - No special licensing is necessary.
- Configuring
 - Configuration file is `DIVA\Program\conf\dfm\dfm.conf`.
- Service Starting
 - `DIVA_HOME\Program\InterLink\dfm\bin\dfm.bat start`
 - The DFM service can also be started using the Windows Services Panel.
- Service Stopping
 - `DIVA\Program\InterLink\dfm\bin\dfm.bat stop`
 - The DFM service can also be stopped using the Windows Services Panel.
- Service Uninstalling
 - `DIVA_HOME\Program\InterLink\dfm\bin\dfm.bat uninstall`
- Upgrading
 - Use the standard DIVArchive upgrade procedure.
- Service Logging
 - The log folder is `DIVA\Program\log\dfm\`.
 - The logging configuration is in `DIVA\Program\conf\dfm\dfm.trace`.
- Service Monitoring
 - The DFM logs can be used to get the current status.
 - `DIVA_HOME\Program\InterLink\log\dfm\`

6.3 DIVArchive DFM Command Line Interface

DFM commands can be initiated using the command line from the operating system. The command line options are as follows:

Usage:

```
dfm command [options]
```

Where `command` is one of:

- `install`
 - Installs the module as a system service.
- `uninstall`
 - Removes the executable as a system service.

- **start**
 - Starts the module.
- **stop**
 - Stops the module if it is currently running.
- **restart**
 - Stops and subsequently starts the module.
- **status**
 - Display the current status of a system service.
- **version**
 - Displays the module version information and exits.
- **help**
 - Displays this help information and exits.

Where options include:

- **-conf**
 - Identifies the configuration file to load the settings from.
- **-f**
 - A synonym for the **-conf** option.

6.4 Monitoring of the DIVAnet Access Gateway via the DIVArchive Control GUI

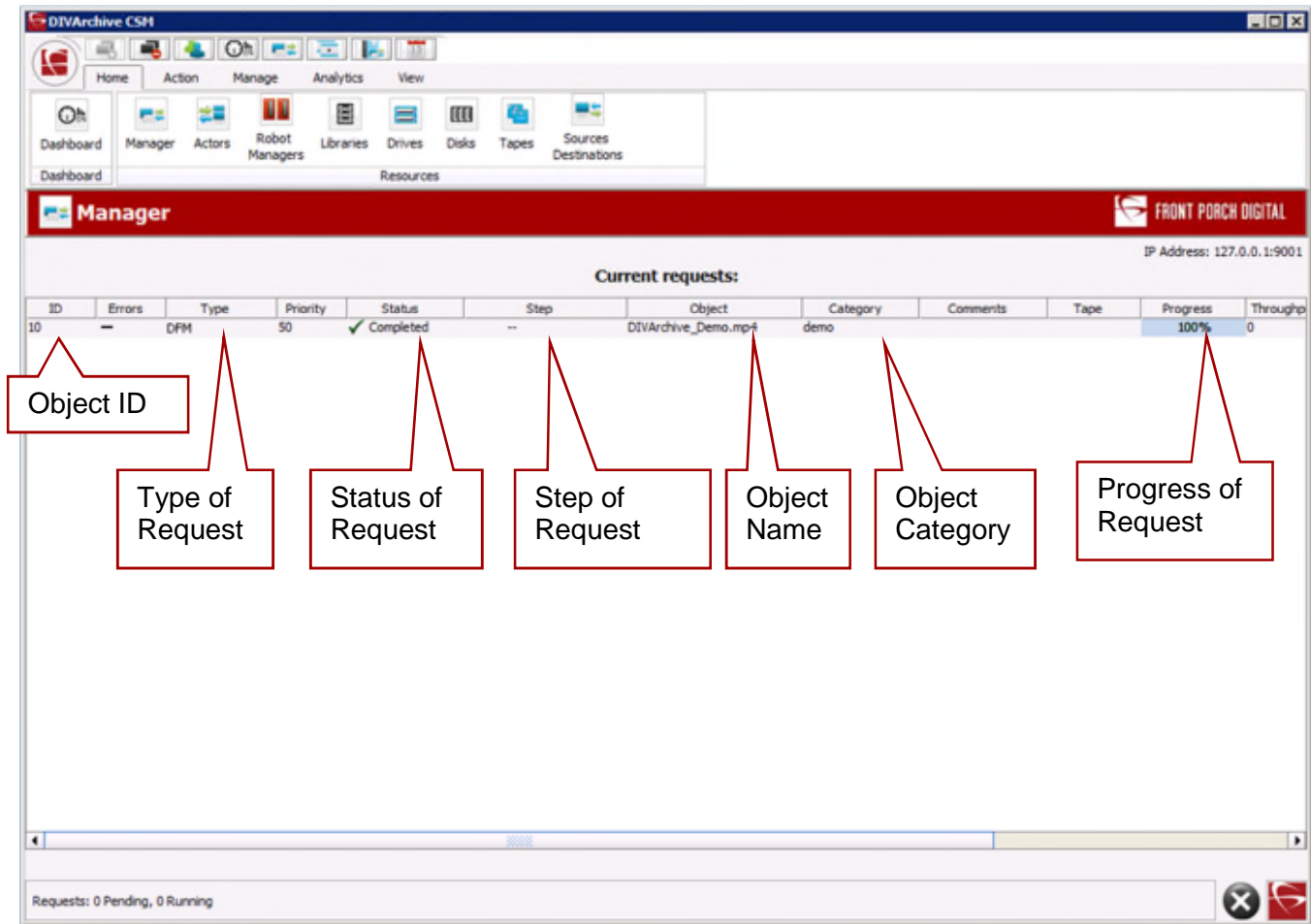
The DIVArchive Control GUI is used for monitoring the requests initiated by DFM. See *A1 DIVArchive Options and Licensing for DIVArchive licensing information*.

To start the Control GUI, use the command:

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

The DIVArchive Control GUI allows viewing the status of the running requests, Actors, and disks. The Objects and Requests panels also offer search functionality.

Figure 7: DIVArchive Control GUI Showing Status DFM Archive Requests



6.5 Monitoring of DFM using Logs

The DFM log files are used for status monitoring of DFM and are located in `DIVA_HOME\Program\InterLink\log\dfm\`.

The different log files are as follows:

- Incomplete Log File (*incomplete.log*)
 - This log file contains information about incomplete files; i.e. files that were put into the folder without a Metadata File, and files that cannot be archived to the system after a specific number of attempts.
 - This file is used to identify incomplete files.
- DIVArchive Request Log File (*request.log*)
 - This log file contains information about all of the requests to DIVArchive and their status; i.e. archive requests performed by the DIVArchive Archive Module and request status obtained by the DIVArchive Status Module.
 - This file is used to obtain all information about requests to DIVArchive and find the root cause of any archive errors.
- Trace Log File (*trace.log*)
 - This log file contains information about all of the files that were found by the File Monitor Module, requests to DIVArchive and their statuses, incomplete files, and any errors.
 - This file is used by the ORACLE Engineering Team.
- Error Log File (*dfm.err*)
 - This log file contains information about all errors that occurred in DFM.
 - This file is used by the administrator to identify the health of DFM.
- Service (*wrapper*) Log File (*service.log*)
 - This log file contains information about the health of the service (*wrapper*).
- Full Trace Log File (*fulltrace.log*)
 - This log file contains all of the information mentioned above.
 - The file is used by the Oracle Engineering Team.
 - See Appendix A6 Sample of the Full Trace Log File for a sample of a Full Trace Log.

6.6 Running Multiple Instances of DFM on the Same Machine

In order to run several instances of DFM on a single machine, only one installation is required but several configuration files will need to be created (*one for each instance*).

The following parameters need to be different for each different DFM instance:

- `<serviceName>`
- `<foibStateFile>`
- `<doibStateFile>`

The following commands are for administration operations:

- Service Installation
 - DFM is installed with DIVArchive in the same installation package.
 - `DIVA_HOME\Program\InterLink\dfm\bin\dfm.bat install -conf configpath\configFile.conf`
- Licensing
 - No special licensing is necessary.
- Configuring
 - The configuration file is `configpath\configFile.conf`.
- Service Starting
 - `DIVA_HOME\Program\InterLink\dfm\bin\dfm.bat start -conf configpath\configFile.conf`
 - The DFM service can also be started using the Windows Services Panel.
- Service Stopping
 - `DIVA_HOME\Program\InterLink\dfm\bin\dfm.bat stop -conf configpath\configFile.conf`
 - The DFM service can also be stopped using Windows Services Panel.
- Service Uninstalling
 - `DIVA_HOME\Program\InterLink\dfm\bin\dfm.bat uninstall -conf configpath\configFile.conf`
- Upgrade
 - Use the standard DIVArchive upgrade procedure.
- Service Logging
 - The log folder is `DIVA_HOME\Program\log\dfm\`.
 - The log configuration is located in `DIVA_HOME\Program\conf\dfm\dfm.trace`.
- Service Monitoring
 - Use the DFM logs to obtain the current status.
 - `DIVA_HOME\Program\InterLink\log\dfm\`

APPENDIX

A1 DIVArchive Options and Licensing

The following table identifies DIVArchive options and licensing metrics.

Part Number	Description	Licensing Metric
L101163	Oracle DIVArchive Nearline Capacity	Per TB
L101164	Oracle DIVArchive Archive Capacity	Per Slot
L101165	Oracle DIVArchive Actor	Per Server
L101166	Oracle DIVArchive Manager	Per Server
L101167	Oracle DIVArchive Partial File Restore	Per Wrapper
L101168	Oracle DIVArchive Avid Connectivity	Per Server
L101169	Oracle DIVArchive Application Filtering	Per Server
L101170	Oracle DIVArchive Storage Plan Manager (2 storage plans are included with a DIVArchive Manager License)	Per Server
L101171	Oracle DIVAnet	Per Server
L101172	Oracle DIVAdirector	Per User
L101918	Oracle DIVArchive Export / Import	Per Server
L101919	Oracle DIVArchive Additional Archive Robotic System	Per Tape Library
L101920	Oracle DIVArchive Automatic Data Migration	Per Server

A2 Object Configuration

All File Set objects are described by a companion MDF (*Metadata*) File. Object configuration in the MDF File should be specified using the following parameters:

Table 6: Object Configuration Parameters

Parameter	Definition
priority	Priority of the DIVArchive request for this object. Object priority can be in the range of [0...100]. The value 0 is the lowest priority and 100 the highest.
categoryName	Object Category Name. This value will be provided to DIVArchive.
objectName	Object Name. This value will be provided to DIVArchive.
comments	Optional information describing the object. This value will be provided to DIVArchive.
sourceDestinationDIVAName	DIVArchive Source/Destination Name used in the archived request.
sourceDestinationDIVAPath	DIVArchive Source/Destination Path used in the archived request.
originalServer	Source/Destination to store in the object catalog, if different from the actual Source/Destination. This is useful when an intermediate storage area was used for archiving but the user wants to store the original source for the data in the object catalog information.
originalPath	File Root Path to store in the object catalog, if different from the actual File Root Path. This is useful when an intermediate storage area was used for archiving but the user wants to store the original source for the data in the object catalog information
fileList	List of File Path Names that should be archived relative to the folder specified in sourceDestinationDIVAPath parameter.

A3 Metadata File

The Metadata File (*MDF*) can be created in the old MDF format or in XML. Initially, DFM will try to parse the MDF as an XML file. If it fails then it will parse it in the old MDF format.

Caution: The last file to be created in a File Set Drop Folder must be the Metadata File (*.mdf*) which then triggers the archive operation. If the Metadata File is transferred BEFORE the actual data files, content loss can occur!

A4 Sample of the *sample.mdf* File in the Old MDF Format

Below is an example of the MDF file in the old format:

```
#
#Object configuration.
#

priority=50
categoryName=category1
objectName=object name
<comments>
  here
  are
  comments
</comments>

sourceDestinationDIVAName=SD1
sourceDestinationDIVAPath=/video1/data/

originalServer=original server
originalPath=original path

<fileList>
  object1/1.mid
  object1/2.mid
  object1/3.mid
</fileList>
```

A5 Example of the *sample.xml* in XML format

Below is an example of the MDF file in XML format:

```
#
#Object configuration.
#

<DIVAObjectDefinition>
  <priority>50</priority>
  <categoryName>category1</categoryName>
  <objectName>object name</objectName>
  <comments>here are our comments</comments>
  <sourceDestinationDIVAName>SD1</sourceDestinationDIVAName>
  <sourceDestinationDIVAPath>/video1/data/</sourceDestinationDIVAPath>
  <originalServer>original_server</originalServer>
  <originalPath>original_path</originalPath>
  <divaMediaName>array1</divaMediaName>
  <fileList>
    <file>object1/1.mid</file>
    <file>object1/2.mid</file>
    <file>object1/3.mid</file>
  </fileList>
</DIVAObjectDefinition>
```

A6 Sample of the Full Trace Log File

Below is an example of the Full Trace Log File for DFM:

```
08/02 15:53:12.263 INFO [main,] TraceManager: Logging (re)configured
with:
file:/E:/app/DIVA/7.0.0.22.0/Program/InterLink/dfm/bin/../../../../conf/dfm
/dfm.trace, found as: file specified in setConfigurationPath()

=====

08/02 15:53:12.283 INFO [main,] TraceManager: Logging (re)configured
with:
file:/E:/app/DIVA/7.0.0.22.0/Program/InterLink/dfm/bin/../../../../conf/dfm
/dfm.trace, found as: file specified in setConfigurationPath()

=====

08/02 15:53:12.465 INFO [Wrapper-Connection,] TraceManager: Logging
(re)configured with:
file:/E:/app/DIVA/7.0.0.22.0/Program/InterLink/dfm/bin/../../../../conf/dfm
/dfm.trace, found as: file specified in setConfigurationPath()

=====

08/02 15:53:12.482 INFO [Wrapper-Connection,] TraceManager: Logging
(re)configured with:
file:/E:/app/DIVA/7.0.0.22.0/Program/InterLink/dfm/bin/../../../../conf/dfm
/dfm.trace, found as: file specified in setConfigurationPath()

=====

08/02 15:53:12.483 INFO [Thread-1,] Start: Starting...

08/02 15:53:12.521 INFO [Thread-1,] TraceManager: Logging (re)configured
with:
file:/E:/app/DIVA/7.0.0.22.0/Program/InterLink/dfm/bin/../../../../conf/dfm
/dfm.trace, found as: file specified in setConfigurationPath()

=====

08/02 15:53:12.869 WARN [Thread-1,] FOInformationBase: The dump file is
absent.

08/02 15:53:12.870 WARN [Thread-1,] FOInformationBase: The dump file is
absent.

08/02 15:53:12.974 INFO [Thread-1,] Main: Drop Folder Monitor started.

08/02 15:53:12.974 INFO [Thread-1,] Start: Started

08/02 15:54:12.970 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 15:55:12.924 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 15:55:12.967 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0
```

08/02 15:56:12.872 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 15:56:12.964 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 15:57:12.869 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 15:57:12.964 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 15:58:12.875 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 15:58:12.967 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 15:59:12.879 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 15:59:12.970 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 16:00:12.881 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 16:00:12.973 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 16:01:12.883 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 16:01:12.976 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 16:02:12.886 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 16:02:12.979 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 16:03:12.889 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 16:03:12.982 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

```

08/02 16:04:12.891 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 16:04:12.985 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 16:05:12.897 DEBUG [Thread-2,] FileMonitorThread: File Discovery
UK - prépa meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/
is accpted!

08/02 16:05:12.988 DEBUG [Thread-4,] FOInformationBase:
getObjectsForArchiving: archiving objects count is 0; max archive
operations is 5; full object list for archive size is 0

08/02 16:05:34.369 INFO [Wrapper-Connection,] Launcher: Stopping DFM...

08/02 16:05:34.408 INFO [Wrapper-Connection,] TraceManager: Logging
(re)configured with:
file:/E:/app/DIVA/7.0.0.22.0/Program/InterLink/dfm/bin/../../../../conf/dfm
/dfm.trace, found as: file specified in setConfigurationPath()

=====

08/02 16:05:34.410 INFO [Wrapper-Connection,] Stop: Stopping Drop Folder
Monitor...

08/02 16:05:34.410 DEBUG [Wrapper-Connection,] DFModuleThread: Stopping
module class com.storagetek.diva.proxy.dfm.FileMonitorThread...

08/02 16:05:34.410 DEBUG [Wrapper-Connection,] DFModuleThread: Module
class com.storagetek.diva.proxy.dfm.FileMonitorThread stopped.

08/02 16:05:34.410 DEBUG [Wrapper-Connection,] DFModuleThread: Stopping
module class com.storagetek.diva.proxy.dfm.DMFileMonitorThread...

08/02 16:05:34.410 DEBUG [Wrapper-Connection,] DFModuleThread: Module
class com.storagetek.diva.proxy.dfm.DMFileMonitorThread stopped.

08/02 16:05:34.410 DEBUG [Wrapper-Connection,] DFModuleThread: Stopping
module class com.storagetek.diva.proxy.dfm.DIVAArchiveThread...

08/02 16:05:34.410 DEBUG [Wrapper-Connection,] DFModuleThread: Module
class com.storagetek.diva.proxy.dfm.DIVAArchiveThread stopped.

08/02 16:05:34.410 DEBUG [Wrapper-Connection,] DFModuleThread: Stopping
module class com.storagetek.diva.proxy.dfm.DIVAStatusThread...

08/02 16:05:34.411 DEBUG [Wrapper-Connection,] DFModuleThread: Module
class com.storagetek.diva.proxy.dfm.DIVAStatusThread stopped.

08/02 16:05:34.411 DEBUG [Wrapper-Connection,] DFModuleThread: Stopping
module class com.storagetek.diva.proxy.dfm.IncompleteLogThread...

08/02 16:05:34.411 DEBUG [Wrapper-Connection,] DFModuleThread: Module
class com.storagetek.diva.proxy.dfm.IncompleteLogThread stopped.

08/02 16:05:34.411 DEBUG [Wrapper-Connection,] DFModuleThread: Stopping
module class
com.storagetek.diva.proxy.dfm.common.config.DFMConfigUpdater...

08/02 16:05:34.411 DEBUG [Wrapper-Connection,] DFModuleThread: Module
class com.storagetek.diva.proxy.dfm.common.config.DFMConfigUpdater
stopped.

```

08/02 16:05:34.411 DEBUG [Wrapper-Connection,] DFModuleThread: Stopping module class com.storagetek.diva.proxy.dfm.DFMStateSaverThread...

08/02 16:05:34.411 DEBUG [Wrapper-Connection,] DFModuleThread: Module class com.storagetek.diva.proxy.dfm.DFMStateSaverThread stopped.

08/02 16:05:34.411 INFO [Wrapper-Connection,] Stop: All Drop Folder Monitor modules stopped. DFM state will be saved.

08/02 16:05:34.414 INFO [Wrapper-Connection,] Stop: Drop Folder Monitor stopped. DFM state was saved.

08/02 16:05:34.414 INFO [Wrapper-Connection,] Launcher: DFM stopped.

08/02 16:05:34.415 INFO [Thread-9,] Stop: Stopping Drop Folder Monitor...

08/02 16:05:34.415 DEBUG [Thread-9,] DFModuleThread: Stopping module class com.storagetek.diva.proxy.dfm.FileMonitorThread...

A7 Sample of Full DFM Configuration File (dfm.conf.ini)

Below is an example of the full configuration file for DFM:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<dfmConfiguration>

  <!-- DIVA Manager address -->
  <managerConnetion>
    <address host="host" port="9000" userName="user" password="password"
      applicationName="DFM"/>
  </managerConnetion>

  <!--
  WARNING: This parameter should be written in one line and there should
  not be any symbols or empty spaces at the beginning of the line.
  Otherwise it will not be correctly parsed by starting script.
  -->
  <serviceName>DIVArchiveDFM</serviceName>

  <!-- File name of the File/Object Information Base dump -->
  <foibStateFile>../../program/interlink/dfm/bin/foibState.dmp</foibStateF
  ile>

  <!-- File name of the Delete File/Object Information Base dump -->
  <doibStateFile>../../program/interlink/dfm/bin/doibState.dmp</doibStateF
  ile>

  <!-- File reloading threshold in seconds. This interval will specify
  how
    often File Monitor Module will reload state of DFM folders.
    It should be positive integer value.
  -->
  <fileReloadingThreshold>60</fileReloadingThreshold>

  <!-- Time interval, in seconds. If file last modification time is not
    changed after this interval it will be marked as "copied" and
    archived. It should be positive integer value.
  -->
  <fileReadyThreshold>10</fileReadyThreshold>

  <!-- Directory delete threshold in seconds(for folders in DELETE mode
    only). This interval will specify how often DFM File Monitor
    Module will reconnect to storage areas to reload state of DFM
    folders. If the directory is empty, it will be deleted upon the
    second connection. It should be positive integer value.
  -->
  <dirDeleteThreshold>600</dirDeleteThreshold>

  <!-- File delete threshold in seconds (for folders in DELETE mode
    only). This interval will specify when file will be deleted from
    DFM folder after file copying to folder is completed. The state
    of completion will be identified by the file size. It should be
    positive integer value.
  -->
```

```

<fileDeleteThreshold>86400</fileDeleteThreshold>

<!-- Maximum number of concurrent archive operations that can be
      performed by DFM simultaneously. The rest objects will wait until
      some archive operations will be completed.
-->
<maxArchiveOperations>5</maxArchiveOperations>

<!-- Maximum number of re-archive operations that can be performed by
      DFM if DIVA rejected at request initializing. If this limit will
      be reached then the object will be marked as "incomplete".
-->
<maxRejectCountInitializing>100</maxRejectCountInitializing>

<!-- Maximum number of re-archive operations that can be performed by
      DFM if DIVA rejected on the request processing. If this limit
      will be reached then the object will be marked as "incomplete".
-->
<maxRejectCountProcessing>100</maxRejectCountProcessing>

<!-- Threshold before DIVA Archive Module re-request DIVA after
      rejection. This parameter should be provided in seconds.
-->
<repeatRequestDIVAThreshold>180</repeatRequestDIVAThreshold>

<!-- This parameter specify polling interval of request state for DIVA
      Status Module. This value should be specified in seconds.
-->
<pollingIntervalOfRequestState>30</pollingIntervalOfRequestState>

<!--
=====
      Default folder configurations.
      They may be overridden in folder-specific section
=====
-->
<defaultFolderConfig>
  <!-- There will be two types of directories: single and file set. If
        folder should be single - "single" value should be specified,
        if folder should be file set - "set" value should be
        specified.

        WARNING. Usage of the "Single" mode in combination with the FTP
        server source is not recommended. If the network connection is
        lost partial file may be stored in the archive system.
  -->
  <type>single</type>
  <!-- DFM folder work mode. Can be Archive or Delete. -->
  <mode>Archive</mode>

  <!-- Default priority of the requests for the folder. Folder priority
        can be in the range [0..100]. The value 0 is the lowest
        priority and 100 the highest.
  -->
  <priority>30</priority>

```



```

<!-- Default value for category name. This value will be provided to
      DIVA if it will not be overridden in folder-specific
      configuration.
-->
<categoryName>dfmCategory</categoryName>

<!-- If file is copied in "file set" folder and during
      incompleteThreshold seconds it was not archived it will be
      marked as "incomplete".
-->
<incompleteThreshold>86400</incompleteThreshold>

<!-- Object Name Convert Rule. Only "simple" algorithm of conversion
      is implemented. -->
<objectNameConvertRule
      method="simple">Name.Ext</objectNameConvertRule>

<!-- Option to configure extension for the meta data file for each
      folder. -->
<mdfExtension>mdf</mdfExtension>

<!-- DIVA Original Server used in the archived request. -->
<originalServer>Original Server</originalServer>

<!-- DIVA Original Path used in the archived request. -->
<originalPath>Original Path</originalPath>

<!-- DIVA Source Destination name used in the archived request. -->
<sourceDestinationDIVAName>SourceDestinationServer</sourceDestinationDIV
AName>

<!-- This field defines location of the Drop Folder root on the DIVA
      Source Destination. In case of dedicated source destination (so
      root of S/D will point to the root of DFM), FPR of S/D may
      contain a path and this field may be left empty.
-->

<sourceDestinationDIVAPath>SourceDestinationPath</sourceDestinationDIVAP
ath>

<!-- Incomplete folder/file processing strategy. The value could be
      None, Delete, Rename
-->
<incompleteProcessingStrategy>Rename</incompleteProcessingStrategy>

<!-- Defines threshold on number of files in the same directory. If
      the number of files in the directory will be large then a
      numberFilesArchiveWildcard, DFM will send the archive request
      to manager using the wildcard "*" for the whole directory. The
      value is integer.
-->
<numberFilesToArchiveWildcard>10</numberFilesToArchiveWildcard>

<!-- This parameter is applicable only to "single" drop folders. Using
      this parameter one can specify the rule for which the Root Path
      of the Archive Request will be generated.
      - platform is "WIN" | "SOL" | "CIFS" | "DETECT"

```

```

    The format of the value will be combination of directory names,
    separators, keywords PARENTn, URL_TO_FILE, URL_TO_PARENTn.
    Keywords that are supported:
    1) PARENTn - the directory name, where the definition is:
        DFM_FILE_FULL_PATH = DFM_URL \ PARENTn ... \ PARENT 3 \ PARENT
        2 \ PARENT 1 \ filename.ext
    2) URL_TO_PARENTm - the path from DFM_URL to PARENTm.
    3) URL_TO_FILE - the path from DFM_URL to file.
-->
<archiveFilePathTemplate platform="DETECT"
    options="">URL_TO_FILE</archiveFilePathTemplate>

<!-- This parameter is applicable only to "single" drop folders. Using
    this parameter one can specify the rule for which the file
    names of the Archive Request will be generated.

    - platform is "WIN" | "SOL" | "CIFS" | "DETECT"

    The format of the value will be combination of directory names,
    separators, keywords PARENTn, URL_TO_FILE, filename.ext.
    Keywords that are supported:
    1) PARENTn - the directory name, where the definition is:
        DFM_FILE_FULL_PATH = DFM_URL \ PARENTn ... \ PARENT 3 \
        PARENT 2 \ PARENT 1 \ filename.ext
    2) URL_TO_FILE - the path from DFM_URL to file.
    3) filename.ext - original file name.
    4) filename - file name without extension.
-->
<archiveFileNameTemplate platform="DETECT"
    options="">filename.ext</archiveFileNameTemplate>

<!-- This parameter specify if DFM should Delete Object before Archive
    Request or not for specific folder.
    This parameter can have value TRUE or FALSE.
    This is optional parameter. Default vale is FALSE.
-->
<deleteBeforeArchive>FALSE</deleteBeforeArchive>

<!-- This parameter specify if DFM should Delete child and parent
    folders of MDF file.
    This parameter can have value TRUE or FALSE.
    This is optional parameter. Default vale is TRUE.
-->

<deleteParentDirectoryAndContentDirectories>TRUE</deleteParentDirectoryA
ndContentDirectories>

</defaultFolderConfig>

<!--
=====
    Folders configurations.
=====
-->
<folders>

```

```

<folderConfig>
  <!-- Folder URL. -->
  <url>ftp://diva:diva@localhost:21/dropfolder1</url>

  <!-- There will be two types of directories: single and file set. If
        folder should be single - "single" value should be specified,
        if folder should be file set - "set" value should be
        specified.
        -->
  <type>single</type>

  <!-- Folder priority can be in the range [0..100]. The value 0 is the
        lowest priority and 100 the highest.
        -->
  <priority>30</priority>

  <!-- This parameter can have Primary or Secondary values. Primary -
        configuration parameters specified in mdf are used in archive
        request. Secondary - configuration parameters specified in
        folder specific configuration are used in archive request.
        Default valued, used when this tag is empty or missing -
        Primary.
        -->
  <mdfConfigPriority>Primary</mdfConfigPriority>

  <!-- Folder category name. This value will be provided to DIVA. -->
  <categoryName>Category1</categoryName>

  <!-- If file is copied in "file set" folder and during
        incompleteThreshold seconds it was not archived it will be
        marked as "incomplete".
        -->
  <incompleteThreshold>86400</incompleteThreshold>
  <!-- DIVA Source Destination name used in the archived request. -->
  <sourceDestinationDIVAName>FTP</sourceDestinationDIVAName>

  <!-- This parameter is applicable only to "single" drop folders.
        Using this parameter one can specify the rule for which the
        Root Path of the Archive Request will be generated.

        - platform is "WIN" | "SOL" | "CIFS" | "DETECT"

        The format of the value will be combination of directory names,
        separators, keywords PARENTn, URL_TO_FILE, URL_TO_PARENTn. Keywords that
        are supported:
        1) PARENTn - the directory name, where the definition is:
            DFM_FILE_FULL_PATH = DFM_URL \ PARENTn ... \ PARENT 3 \
            PARENT 2 \ PARENT 1 \ filename.ext
        2) URL_TO_PARENTm - the path from DFM_URL to PARENTm.
        3) URL_TO_FILE - the path from DFM_URL to file.
        -->
  <archiveFilePathTemplate platform="DETECT"
options="">URL_TO_FILE</archiveFilePathTemplate>

```

```

<!-- This parameter is applicable only to "single" drop folders.
      Using this parameter one can specify the rule for which the
      file names of the Archive Request will be generated.

- platform is "WIN" | "SOL" | "CIFS" | "DETECT"

The format of the value will be combination of directory names,
separators, keywords PARENTn, URL_TO_FILE, filename.ext.
Keywords that are supported:
1) PARENTn - the directory name, where the definition is:
      DFM_FILE_FULL_PATH = DFM_URL \ PARENTn ... \ PARENT 3 \
      PARENT 2 \ PARENT 1 \ filename.ext
2) URL_TO_FILE - the path from DFM_URL to file.
3) filename.ext - original file name.
4) filename - file name without extension.
-->
<archiveFileNameTemplate platform="DETECT"
  options="">filename.ext</archiveFileNameTemplate>

<!-- DIVA Media Name used in the archived request. -->
<divaMediaName>Array1</divaMediaName>

      <!-- To specify filters <fileFilter> tag is used in folder
      configuration. "type" attribute can be "exclude" or
      "include". In inclusion mode, no files will be archived if no
      filters are defined. Each filter will allow archiving files
      by the specific rule. File will be archived only if file will
      correspond to one or more masks. If file does not correspond
      to any mask, file will be skipped by DFM (nor deleted neither
      archived).

In exclusion mode, all files will be archived if no filters are
defined (the same way is working now). Each filter will
restrict archiving files by the specific rule. File will NOT
be archived if file will correspond to one or more masks. If
file corresponds to any mask, file will be skipped by DFM
(nor deleted neither archived).

Filter can be defined as a pattern (mask which can have ? which
means any symbol or * which means any string) or as a Regex
expression (please see details
http://en.wikipedia.org/wiki/Regular\_expression,
http://www.regular-expressions.info).

<mask> specifying pattern for filter.
<regexp> specifying regular expression for filter.

PLEASE NOTE: <fileFilter> parameter is optional for drop
folder configuration. If it is not specified DFM will archive
any file without any restriction except standard DFM logic.

PLEASE NOTE: Only one <fileFilter> can be specified per one
folder.
-->
<fileFilter type="include">

```

```

    <mask>*.*</mask>
</fileFilter>

<!-- This parameter specify if DFM should Delete Object before
      Archive Request or not for specific folder.
      This parameter can have value TRUE or FALSE.
      This is optional parameter. Default vale is FALSE.
-->
<deleteBeforeArchive>TRUE</deleteBeforeArchive>

<!-- This parameter specify if DFM should Delete child and parent
      folders of MDF file.
      This parameter can have value TRUE or FALSE.
      This is optional parameter. Default vale is TRUE.
-->

<deleteParentDirectoryAndContentDirectories>FALSE</deleteParentDirectory
AndContentDirectories>

</folderConfig>

<folderConfig>
  <!-- Folder URL. -->
  <url>ftp://diva:diva@localhost:21/dropfolder2</url>

  <!-- There will be two types of directories: single and file set. If
        folder should be single - "single" value should be specified,
        if folder should be file set - "set" value should be
        specified.
  -->
  <type>set</type>

  <!-- Folder priority can be in the range [0..100]. The value 0 is the
        lowest priority and 100 the highest.
  -->
  <priority>30</priority>

  <!-- Folder category name. This value will be provided to DIVA. -->
  <categoryName>Category2</categoryName>

  <!-- If file is copied in "file set" folder and during
        incompleteThreshould seconds it was not archived it will be
        marked as "incomplete".
  -->
  <incompleteThreshold>86400</incompleteThreshold>

  <!-- DIVA Source Destination name used in the archived
        request. -->
  <sourceDestinationDIVAName>FTP</sourceDestinationDIVAName>

  <!-- This field defines location of the Drop Folder root on the DIVA
        Source Destination. In case of dedicated source destination
        (so root of S/D will point to the root of DFM), FPR of S/D
        may contain a path and this field may be left empty.
  -->
  <sourceDestinationDIVAPath>DROPFOLDER</sourceDestinationDIVAPath>

```

```

<!-- DIVA Media Name used in the archived request. -->
<divaMediaName>Array1</divaMediaName>

<!-- DIVA Media Name Pattern used in the archived request. Using this
      pattern GROUP will be obtained -->
<divaMediaNamePattern>def/$GROUP/data</divaMediaNamePattern>

<!-- This parameter should be used to enable recursive archive mode
      of DFM. In this mode DFM will check MDF file and if it
      contains asterisks as part of data files path recursive
      archive will be used.
      This parameter can have value TRUE or FALSE.
      This is optional parameter. Default vale is FALSE.
      -->
<recursiveArchive>TRUE</recursiveArchive>

</folderConfig>

</folders>

<!--WARNING: This is not a comment. Do not modify the text below. It is
      used by Tanuki Wrapper.
*****
#Wrapper Properties
*****
# Java Application
wrapper.java.command=%JAVA_HOME%\bin\javaw.exe

# Java Main class. This class must implement the WrapperListener
interface
# or guarantee that the WrapperManager class is initialized. Helper
# classes are provided to do this for you. See the Integration section
# of the documentation for details.
wrapper.java.mainclass=com.storagetek.diva.proxy.dfm.control.TanukiLaunc
her

# Java Classpath (include wrapper.jar) Add class path elements as
# needed starting from 1
wrapper.java.classpath.1=../../Program/InterLink/lib/Interlink.jar
wrapper.java.classpath.2=../wrapper.jar
wrapper.java.classpath.3=../../Program/Common/lib/Common.jar
wrapper.java.classpath.4=../log4j/log4j.jar
wrapper.java.classpath.5=../JavaLib/xerces.jar
wrapper.java.classpath.6=../JavaLib/dtdparser115.jar
wrapper.java.classpath.7=../Apache/commons-io-1.3.2.jar

#wrapper.java.classpath.7=../libs/jhall.jar
#wrapper.java.classpath.8=../libs/junit.jar
#wrapper.java.classpath.9=../libs/ojdbc14.jar

# Java Library Path (location of Wrapper.DLL or libwrapper.so)
wrapper.java.library.path.=.

# Java Additional Parameters

```

```

#wrapper.java.additional.1=
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=64

# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=256

# Application parameters. Add parameters as needed starting from 1

#####
# Wrapper Logging Properties
#####
# Format of output for the console. (See docs for formats)
wrapper.console.format=PM

# Log Level for console output. (See docs for log levels)
wrapper.console.loglevel=INFO

# Format of output for the log file. (See docs for formats)
wrapper.logfile.format=LPTM

# Log Level for log file output. (See docs for log levels)
wrapper.logfile.loglevel=INFO

# Maximum size that the log file will be allowed to grow to before
# the log is rolled. Size is specified in bytes. The default value
# of 0, disables log rolling. May abbreviate with the 'k' (kb) or
# 'm' (mb) suffix. For example: 10m = 10 megabytes.
wrapper.logfile.maxsize=10m

# Maximum number of rolled log files which will be allowed before old
# files are deleted. The default value of 0 implies no limit.
wrapper.logfile.maxfiles=4

# Log Level for sys/event log output. (See docs for log levels)
wrapper.syslog.loglevel=NONE

#####
# Wrapper Windows Properties
#####
# Title to use when running as a console
wrapper.console.title=DIVArchive DFM

#####
# Wrapper Windows NT/2000/XP Service Properties
#####
# WARNING - Do not modify any of these properties when an application
# using this configuration file has been installed as a service.
# Please uninstall the service before modifying this section. The
# service can then be reinstalled.

# Time without CPU before JVM will issue warning and extend timeout (in
# sec).
# Timeout will be extended by a few seconds at least once before Wrapper
# shuts down.

```

```
#wrapper.cpu.timeout=30

# Number of seconds to allow between the time that the Wrapper launches
the JVM process and the time that the JVM side of the Wrapper responds
that the application has started.
wrapper.startup.timeout=60

# Number of seconds to allow between the wrapper pinging the JVM and the
response
wrapper.ping.timeout=60

# Number of seconds to allow between the time that the Wrapper asks the
JVM to shutdown and the time that the JVM side of the Wrapper responds
that it is stopping.
wrapper.shutdown.timeout=60

# Name of the service
wrapper.ntservice.name=dfm

# Display name of the service
wrapper.ntservice.displayname=DIVArchive DFM
# Description of the service
wrapper.ntservice.description=DIVArchive DFM allows to monitor FTP and
local directories and to archive incoming file to the DIVA system

# Service dependencies. Add dependencies as needed starting from 1
wrapper.ntservice.dependency.1=

# Mode in which the service is installed. AUTO_START or DEMAND_START
wrapper.ntservice.starttype=AUTO_START

# Allow the service to interact with the desktop.
wrapper.ntservice.interactive=false

-->
</dfmConfiguration>
```