



DIVA

Watch Folder Monitor User Guide

Release: 9.0

Revision: 1.0

Copyrights and Trademark Notices

Specifications subject to change without notice. Copyright © 2024 Telestream, LLC and its Affiliates. Telestream, CaptionMaker, Cerify, DIVA, DIVA, Episode, Flip4Mac, FlipFactory, Flip Player, Gameshow, GraphicsFactory, Kumulate, Lightspeed, MetaFlip, Post Producer, Prism, ScreenFlow, Split-and-Stitch, Switch, Tempo, TrafficManager, Vantage, VOD Producer, and Wirecast are registered trademarks and Aurora, ContentAgent, Cricket, e-Captioning, Inspector, iQ, iVMS, iVMS ASM, MacCaption, Pipeline, Sentry, Surveyor, Vantage Cloud Port, CaptureVU, Cerify, FlexVU, PRISM, Sentry, Stay Genlock, Aurora, and Vidchecker are trademarks of Telestream, LLC and its Affiliates. All other trademarks are the property of their respective owners.

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.

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.

Contents

Telestream Contact Information 6

Preface 7

- Audience 8
- Documentation Accessibility 8
- Related Documents 8
- Document Updates 8

Overview 9

- Overview 10
- Features 10
 - Single File Mode Watch Folders 11
 - File Set Watch Folders 11
- Common Watch Folder Monitor Solutions 12
 - Basic Solution for Single Mode Folders 12
 - Basic Solution for File Set Mode Folders 13
 - Basic Solution for Delete Mode 13
 - Basic Solution for DIVA Connect 13

Solutions and Basic Configurations 15

- Basic Single Mode Folder Solution 16
 - Configuring the Basic Single Mode Folder Solution 17
- Solution for File Set Folders on Local Disk and FTP Servers 18
 - Configuring Recursive Archive 19
 - Configuring the File Set Folders Solution 20
- Delete Mode Solution 21
 - Configuring the Delete Mode Solution 22
- DIVA Connect Solution 23
 - Configuring the DIVA Connect Solution 23
- Running Multiple WFM Instances on the Same Computer 25

Additional Configuration Options 27

Additional WFM Configuration	28
Incomplete File Threshold: <incompleteThreshold>	28
Processing of Incomplete Files: <incompleteProcessingStrategy>	28
MDF Configuration Priority: <mdfConfigPriority>	29
Additional Operations WFM Configuration	29
Delete Before Archive: <deleteBeforeArchive>	29
Number of Files to Archive Wildcard: <numberOfFilesToArchiveWildcard>	29
Object Name Conversion Rules: <objectNameConvertRules>	30
Original Server: <originalServer>	31
Original Path: <originalPath>	31
Media Name: <divaMediaName> and <divaMediaNamePattern>	31
Additional Monitored Folder Configuration	32
MDF File Extension: <mdfExtension>	32
Server Path: <sourceDestinationDIVAPath>	32
Sample Server Paths for Different Configurations	32
Archive File Path Template: <archiveFilePathTemplate>	34
Archive File Name Template: <archiveFileNameTemplate>	34
File Filter: <fileFilter>	35
Delete Parent and Content Directories: <deleteParentDirectoryAndContentDirectories>	36
Folder Configuration Examples	37
Example of a Local Folder in File Set Mode	37
Example of an FTP Folder in Single File Mode	38
Example of a CIFS Folder in File Set Mode	39
Service Log Configuration	40
Trace Log Configuration	41
Advance WFM Configuration	42
Full WFM Configuration File	42
Global WFM Configuration	42
Folders Default Configuration	44
Folder Specific Configuration	46
File Root Path and Single File Mode Advanced Configuration	47

Administration, Operations and Monitoring 51

Starting, Stopping, and Restarting Watch Folder Monitor Overview	52
Starting and Restarting the Watch Folder Monitor Service	52
Running the Watch Folder Monitor Service	52
Stopping the Watch Folder Monitor Service	53
Summary of WFM Administration	53
Command-Line Interface	54
Operations and Workflows	55
Algorithm and Workflow for Single File Mode Watch Folders	56
Algorithm and Workflow for File Set Mode Watch Folders	57
Monitoring Watch Folder Monitor	59
Monitoring DIVA Connect from the Web App	59

Monitoring WFM using Logs 60

Configuration Files and Examples 61

Configuring Objects 62

Metadata File 63

 Sample of the sample.mdf File in XML Format 63

 Sample of the sample.mdf File in Original MDF Format 63

Sample of the Full Trace Log File 64

Full Watch Folder Monitor Configuration File (dfm.conf.ini) 67

Telestream Contact Information

To obtain product information, technical support, or provide comments on this guide, contact us using our web site, email, or phone number as listed below.

Resource	Contact Information
DIVA Technical Support	<p>Web Site: https://www.telestream.net/telestream-support/</p> <p>Depending on the problem severity, we will respond to your request within 24 business hours. For P1, we will respond within 1 hour. Please see the Maintenance & Support Guide for these definitions.</p> <ul style="list-style-type: none"> • Support hours for customers are Monday - Friday, 7am - 6pm local time. • P1 issues for customers are 24/7.
Telestream, LLC	<p>Web Site: www.telestream.net</p> <p>Sales and Marketing Email: info@telestream.net</p> <p>Telestream, LLC 848 Gold Flat Road, Suite 1 Nevada City, CA USA 95959</p>
International Distributor Support	<p>Web Site: www.telestream.net</p> <p>See the Telestream Web site for your regional authorized Telestream distributor.</p>
Telestream Technical Writers	<p>Email: techwriter@telestream.net</p> <p>Share comments about this or other Telestream documents.</p>

Preface

This book describes the DIVA Watch Folder Monitor solutions, configuration, administration, and operations.

Topics

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Document Updates](#)

Audience

This document is intended for system administrators and DIVA users.

Documentation Accessibility

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

Related Documents

For more information, see the DIVA library located at:

<https://www.telestream.net/telestream-support/>

Document Updates

The following table identifies updates made to this document.

Date	Update
April 2022	Updated Copyright information. Updated book for release 8.2. Updated terminology to new standards.
June 2022	Migrated book to Telestream format.
August 2022	Minor terminology and formatting updates.
September 2022	Updated terminology and title page graphic. (see the Overview for updated terms)
October 2022	Updated book for release 8.3.
November 2022	Reverted the term “Virtual Object” to “Object”.
December 2022	Updated book for 8.3.1 release.
February 2023	Updated book for 9.0 release. Updated copyright dates.
March 2023	Change prod name from DIVA Core to Content Manager
September 2023	Change prod name from Content Manager to DIVA. Produce version 9.0 PDF.

Overview

This chapter describes an overview of the WFM (Watch Folder Monitor) component of the DIVA system and includes the following information:

Topics

- [Overview](#)
- [Features](#)
- [Common Watch Folder Monitor Solutions](#)

See the DIVA Architecture, Concepts and Glossary book for more information on DIVA.

Overview

Watch folder monitoring allows users and third party applications to deliver content to be archived by copying related files to a folder, an FTP server, or a CIFS share.

Note: Linux-based Actors do not support UNC paths for CIFS Source Servers and targets.

The Watch Folder Monitor does not require an application, or user, to send the Archive command. Each time a file (Single File Mode), or set of files (File Set Mode), is placed in a monitored watch folder, DIVA automatically archives the files and creates the related object according to the rules set in the configuration file for that particular watch folder.

The flexibility of enables a myriad of possible configuration combinations. Telestream has combined the most commonly used configurations into four basic Watch Folder Monitor solutions to make configuration and operations easier. If necessary, advanced configuration can be performed to employ beyond the four solutions described in this book.

Watch Folder Monitor supports multiple automated operations. Example configurations have been included to aid in the configuration of specific workflows as follows:

- wfm.conf.delete.ini

This configuration file contains the minimum configuration parameters to configure for use as a file deletion solution.

- wfm.conf.single.local.ini

This configuration file is a template for configuring automated local, single file based archive operations.

- wfm.conf.ini

This configuration file is a template demonstrating how multiple workflows and watch folders can coexist in a single configuration file, and is modular by design.

Features

Watch Folder Monitor is a component of DIVA that can be installed with the DIVA installation package.

Note: In DIVA, Watch Folder Manager is still named DFM (Drop Folder Manager), which was the legacy name. The configuration files and command line interfaces were not renamed and are still called *dfm*.

WFM monitors designated watch folders and requests DIVA to archive objects found in these folders. WFM can monitor up to 20 local folders, FTP folders, and CIFS folders;

each with its own individual configuration. There are two types (modes) of watch folders; Single File Mode and File Set Mode.

For example, if the archive contains two media Tape Groups, News and Movies, then two watch folders must be configured; one for News and one for Movies. However, if File Set Mode is used (or an advanced configuration setup), a single watch folder can be used instead.

After archiving completes, the files are deleted from the folder they have been archived from.

Best practices dictate that files should be copied to the watch folder and not moved. After the configured time period, if the file is considered incomplete, it might be deleted. Copying the file, rather than moving it, ensures that a backup copy is available if an unwanted deletion occurs.

When the watch folder is configured on an FTP server, all content (files and folders to be archived) must be copied to the watch folder using FTP. If the content is copied to the watch folder using another process (local copy through the operating system, remote desktop copy, and so on), there is no guarantee it will process the content properly.

Single File Mode Watch Folders

Use Single File Mode Watch Folders to automatically archive objects containing a single file. This mode is convenient for MXF, GXF, and so on, 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 file name including the extension. The Object Collection, Priority, and other archive command parameters are defined in a watch folder configuration file (`wfm.conf`). The configuration file contains the description of all folders to be monitored by WFM.

When a file appears in a configured watch folder, WFM monitors the file size. If the size has not changed after a preconfigured time threshold (last modification time), then the file is considered complete, and the archive is initiated. After the file is archived it is automatically deleted from the watch folder.

File Set Watch Folders

Caution: The last file created in the File Set Watch Folder must be the metadata file (with the .mdf extension). The presence of this file triggers the archive operation. If the metadata file is transferred before the actual data files, content loss can occur!

File Set Watch Folders are used to automatically archive objects containing more than one file, and are compatible with complex objects.

When monitoring a File Set Watch Folder, WFM checks for the presence of a metadata file (.mdf). This file describes the details of the content to be archived, and signals WFM

that the content to be archived is ready. The archive process starts only after this file is available to WFM.

Using the metadata file allows specifying the Object Name, Object Collection and user Comments on a per object basis.

Users (or an application) must first create a folder within the watch folder for the set of files intended to be archived as a single object in DIVA. The folder must have the `-allow_delete_on_source` and `-r` options configured. After the folder is created, the data files have to be copied to the folder.

The object archiving process is initiated after WFM has opened and parsed the metadata file. After the object is successfully archived, WFM deletes the folder, and all of the files within it, including the `.mdf` file.

Common Watch Folder Monitor Solutions

Telestream offers the four most common Watch Folder Monitor solutions as described in the following sections. Select the solution, or solutions, that meet the requirements of the system environment, and follow the configuration instructions for those particular solutions.

See [Solutions and Basic Configurations](#) for details and basic configuration of each solution.

Basic Solution for Single Mode Folders

This solution can monitor local disks, FTP transfers, and CIFS folders. DIVA will archive each file found in the watch folder as a single object. If only single files are required then this solution will work perfectly.

- Only single file objects are supported.
- Supported major configuration parameters include:
 - DIVA
 - Watch folder URL
 - Collection Name of objects being archived
 - Server for the Archive job
 - Templates for File Path Root and File Name generation
 - Media Name for the Archive job

If the connection to the disk, FTP or CIFS is unavailable (or disconnected) for a period (that is, 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 `wfm.conf.ini`, and not in the `wfm.single.conf.ini` configuration file. However, this parameter can be manually added to the `wfm.single.conf.ini` file if wanted, or necessary.

Basic Solution for File Set Mode Folders

This solution offers more flexibility than the basic solution. This configuration can monitor local disks, FTP servers, and unsecured CIFS folders containing either single files or file sets. Secured CIFS can also be used if the service is run using a suitable user account.

File Sets include a metadata file containing the details of the files included in the set. Each metadata file found in the watch folder is parsed and a new object (usually with more than one component) is archived in the DIVA system.

- Single and multiple file objects are supported.
- Recursive Archive (folder and files) is supported.
- Supported major configuration parameters include:
 - DIVA
 - Watch folder URL
 - Media Name for the Archive job

Basic Solution for Delete Mode

This 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 with the other WFM solutions, and can be included in all WFM configurations.

Supported major configuration parameters include:

- Folder URL
- Interval specifying when the file will be deleted from the folder.

Basic Solution for DIVA Connect

This solution monitors only the local disk and can use either Single Files or File Sets. Each metadata file found in the watch folder is parsed and a new object (usually with more than one component) is archived in the DIVA System.

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

- Single and multiple file objects are supported
- Recursive Archive (folders and files) is supported

- Supported major configuration parameters include:
 - DIVA
 - Watch folder URL
 - Media Name Pattern for generation of the Media Name for the Archive job
 - Server Name
 - Server Path

Solutions and Basic Configurations

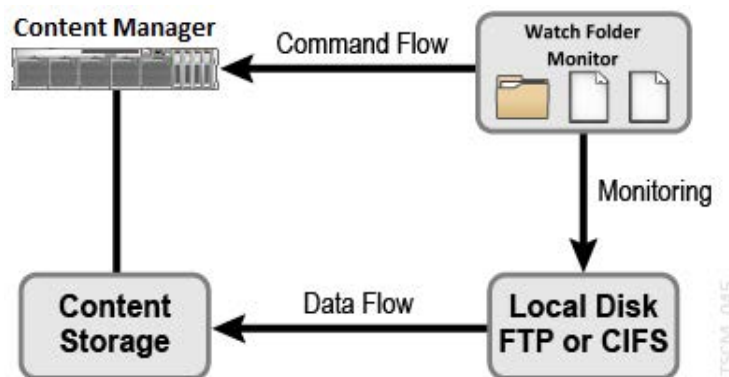
This chapter describes details and configurations of the different Watch Folder Monitor solutions, and includes the following information:

Topics

- [Basic Single Mode Folder Solution](#)
- [Solution for File Set Folders on Local Disk and FTP Servers](#)
- [Delete Mode Solution](#)
- [Content Conductor Connect Solution](#)
- [Running Multiple WFM Instances on the Same Computer](#)

Basic Single Mode Folder Solution

This solution can monitor a local disk, FTP transfers, and CIFS folders. DIVA will archive each file found in the watch folder as a single object. As files are dropped into the folder they are archived as single files into DIVA. Multiple files can be in the watch folder at the same time, but they will be handled by the Watch Folder Monitor one at a time.



Supported Servers include:

- Local disks
- FTP folders

When the watch folder is configured on an FTP server, all content (files and folders to be archived) must be copied to the watch folder using FTP. If the content is copied to the watch folder using another process (local copy through the operating system, remote desktop copy, and so on), there is no guarantee WFM will process the content properly.

- CIFS folders

Linux-based Actors do not support UNC paths for CIFS Source and Target Servers.

The following are additional configuration parameters and functionality:

- Only single file objects are supported.
- Supported major configuration parameters include:
 - DIVA
 - Watch folder URL
 - Collection Name of objects being archived
 - Server for the Archive job
 - Templates for File Path Root and File Name generation
 - Media Name for the Archive job

Configuring the Basic Single Mode Folder Solution

In the basic WFM solution, DIVA will archive each file found in the watch folder as single object. Use the following procedure to configure this solution for a Single Mode Folder. Editing the WFM configuration file is necessary to make the following changes.

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

The file is located in the `%DIVA_HOME%\Program\conf\dfm\` folder.

2. Specify the Host and Port using `<managerConnetion>` as follows (using the proper parameters):

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

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

```
<serviceName>diva_dfm</serviceName>
```

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

4. Specify the Folder URL using `<url>` as follows (using the proper URL):

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

5. Specify the Collection Name of objects to be archived using `<categoryName>` as follows (using the proper Collection Name):

```
<categoryName>Collection</categoryName>
```

6. Specify the Servers to use in Archive requests using

`<sourceDestinationDIVAName>` as follows using the proper Server name:

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

7. Specify the File Path to the folder containing the files DIVA will archive using `<archiveFilePathTemplate>` as follows (using the proper File Path):

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

8. Specify the template for the generation of the File Name using

`<archiveFileNameTemplate>` as follows (using the proper Template Name):

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

9. Specify the Media Name used in the archive request using `<divaMediaName>` as follows (using the proper Media Name):

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

Caution: If the following Step 10 is not completed, the installation will fail because it cannot find the trace file.

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

11. Install the WFM service using the following command:

```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat install
```

12. Start the WFM service using the following command, or open the Windows Services panel and start the service from there.

```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat start
```

Solution for File Set Folders on Local Disk and FTP Servers

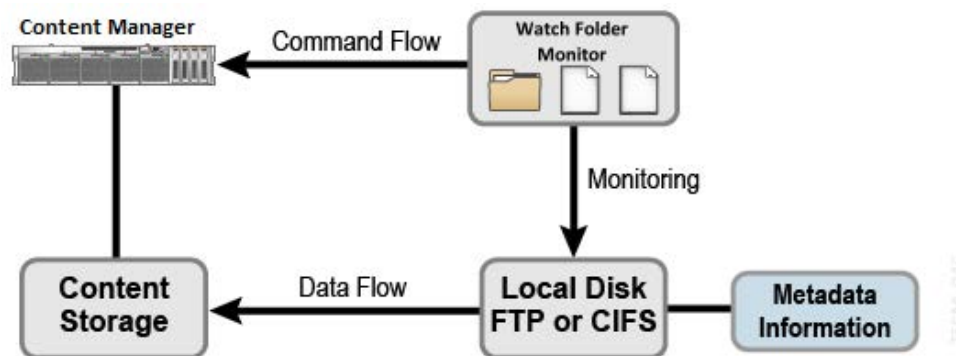
This solution offers more flexibility than the basic solution. This configuration can monitor the local disk, FTP server, and CIFS folders that contain either single files or file sets.

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

Caution: The last file created in the File Set watch folder must be the metadata file (with the .mdf extension). The presence of this file triggers the archive operation. If the metadata file is transferred before the actual data files, content loss can occur!

AXF format files are, at the file system level, single files, but contain many files internally. AXF files can be archived into DIVA, but this requires the use of a File Set mode folder and an MDF file. The MDF file contains the list of files within the AXF file, and the target parent attributes of the object used when archived.

See [Configuration Files and Examples](#) for sample metadata files in WFM format.



Supported Servers include:

- Local disks
- FTP folders

When the watch folder is configured on an FTP server, all content (files and folders to be archived) must be copied to the watch folder using FTP. If the content is copied to the watch folder using another process (local copy through the operating system, remote desktop copy, and so on), there is no guarantee WFM will process the content properly.

- Secured CIFS folders

Secured CIFS is possible if the Windows WFM Service is run under a suitable user account.

The following are additional configuration parameters and functionality:

- Single and multiple file objects are supported.
- Recursive Archive (folders and files) is supported
- Supported major configuration parameters include:
 - DIVA
 - Watch folder URL
 - Media Name for the Archive job

Configuring Recursive Archive

Recursive Archive is enabled when the WFM folder is configured to use File Set Mode. In this mode the Archive File Set is defined in the specific object configuration .mdf file.

This feature defines a special character, an asterisk, that can be used in the WFM metadata file (the .mdf file). Applying an asterisk to the end of the folder path causes all files under this path to be archived, but does not include subfolders.

Using the `-r` option in the configuration of the watch folder achieves different results than using the asterisk option. When the `-r` option is included in the job, the Actor will recurse into all subfolders whether the asterisk is used or not.

Conversely, if the `-r` option is not used, the Actor only selects the files in the immediate folder. In this case, use of the asterisk will still cause archiving of all files in the immediate folder action to occur. When using this option, be aware of the job and server configurations to achieve the desired results.

Configure the `-r` option in either of the two following locations:

- In the job Options field when creating the job. This method is only applicable to the current job. Enter the `-r` option in the Options field to use this method.
- In the Server Connect Options field in the DIVA web app. This method is applicable to all requests to this server. Include the `-r` option in the Connect Options field with any other request options to use this method.

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

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

Example 1:

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

The results of this configuration are as follows:

- Every file under folder1 and test/folder2 will be archived, but not subfolders. If the `-r` option is used in the job Options, or on the Server Configuration, all subfolders and their files are also archived.
- For folder3/test.txt, only the test.txt file located in folder3 will be included.

Example 2:

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

The results of this configuration are as follows:

- Every file in the specified folder will be archived, but not subfolders.
- If the `-r` option is used in the job Options, or on the Server Configuration, all subfolders and their files are also archived.

Example 3:

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

The results of this configuration are as follows:

- Every file under folder1 will be archived, but not subfolders.
- If the `-r` option was used in the job Options, or on the Server Configuration, all folder1 subfolders and their files will also be archived.

Configuring the File Set Folders Solution

Use the following procedure to configure this solution for a File Set folder, default filter for local disks, and FTP:

1. Rename the wfm.conf.set.local.ftp.ini file to wfm.conf.
The file is located in the %DIVA_HOME%\Program\conf\dfm\ folder.
2. Specify the Host and Port using `<managerConnetion>` as follows (using the proper parameters):

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

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

```
<serviceName>diva_dfm</serviceName>
```

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

4. Specify the `<folderConfig>` node using `<type>`. The folder should be the File Set type, and therefore the set value should be specified as follows:

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

5. Specify the Folder URL using `<url>` as follows (using the proper URL):

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

6. Specify the Media Name to use in the archive request using `<divaMediaName>` as follows (using the proper Media Name):

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

Caution: If the following Step 7 is not completed, the installation will fail because it cannot find the trace file.

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

8. Install the WFM service using the following command:

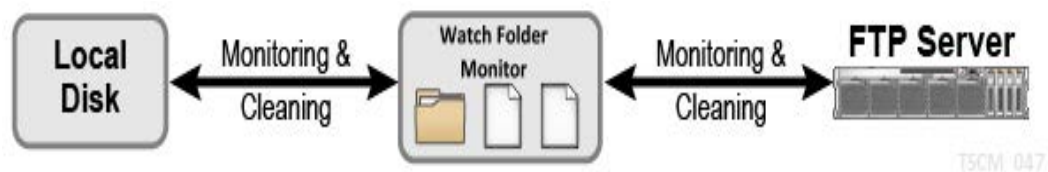
```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat install
```

9. Start the WFM service using the following command, or open the Windows Services panel and start the service from there.

```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat start
```

Delete Mode Solution

This solution monitors the specified folder, and removes obsolete files. If a subfolder is empty for more than the configured allowed time period, it will also be removed. This solution is used in with the other WFM solutions, and should be included in all WFM configurations.



The following major configuration parameters are supported:

- Folder URL
- Interval specifying when files will be deleted from the folder

Configuring the Delete Mode Solution

Use the following procedure to configure the WFM solution for Delete Mode:

1. Rename the `dfm.conf.delete.ini` file to `dfm.conf`.
The file is located in the `%DIVA_HOME%\Program\conf\dfm\` folder.
2. Specify the interval that identifies when the file will be deleted from the folder using `<fileDeleteThreshold>` as follows (the default value is shown):

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

 - This value is the File Delete Threshold in seconds, and for folders only in Delete Mode.
 - This interval specifies when a file will be deleted from a WFM folder after copying the file to the folder is complete.
 - The state of completion is 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 recognizes that the file copy operation is complete.
 - This value must be a positive integer value.
 - The default value of 86400 seconds equals 24 hours.
3. Specify the Windows Service Name using `<serviceName>` as follows:

```
<serviceName>diva_dfm</serviceName>
```

This parameter must be written on one line, and there must not be any symbols or empty spaces at the beginning of the line.
4. Specify the `<folderConfig>` node using `<mode>` as follows:

```
<mode>Delete</mode>
```
5. Specify the Folder URL using `<url>` as follows (using the proper URL):

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

Caution: If the following Step 6 is not completed, the installation will fail because it cannot find the trace file.

6. Rename the `%DIVA_HOME%\Program\conf\dfm\dfm.trace.ini` file to `dfm.trace`.
7. Install the Watch Folder Monitor service using the following command:

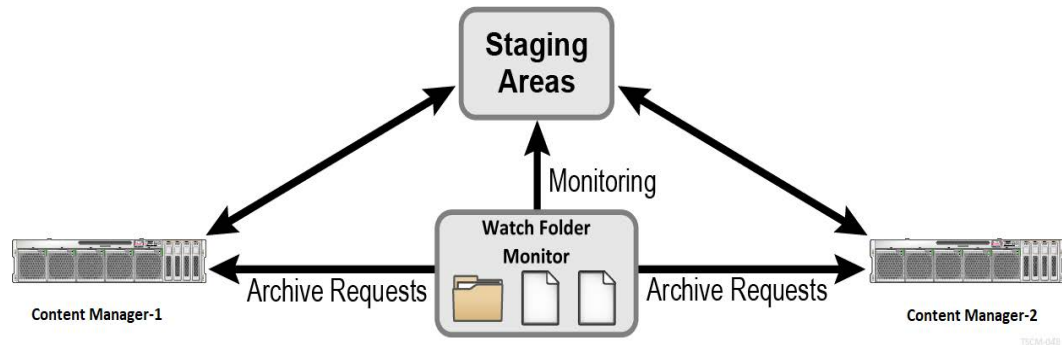
```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat install
```
8. Start the Watch Folder Monitor service using the following command, or open the Windows Services panel and start the service from there.

```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat start
```

DIVA Connect Solution

This solution monitors a local disk, and can use either Single Files or File Sets. Each metadata file found in the watch folder is parsed, and a new object (usually with more than one file) is archived in DIVA. DIVA Connect systems configured with inter-site transfer capabilities use this solution.

Contact Telestream Support for detailed information about various available DIVA Connect solutions.



Supported Servers include:

- Local disks

The following are additional configuration parameters and functionality:

- Single and multiple file objects are supported.
 - Files must be in non-complex objects.
- Supported major configuration parameters include:
 - DIVA
 - Watch folder URL
 - Media Name Pattern for generation of the Media Name for the Archive job
 - Server name
 - Server path

Configuring the DIVA Connect Solution

This solution monitors the local disk. Each metadata file found in the watch folder is parsed, and a new object (usually with more than one file but not a complex object) is archived in DIVA.

Use the following procedure to configure the Watch Folder Monitor Solution for DIVA Connect:

1. Rename the *dfm.conf.DIVA Connect.ini* file to *dfm.conf*.

The file is located in the %DIVA_HOME%\Program\conf\dfm\ folder.

2. Specify the Host and Port using `<managerConnetion>` as follows (using the proper parameters):

```

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

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

```
<serviceName>diva_dfm</serviceName>
```

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

- Specify the `<folderConfig>` node using `<type>`. The folder should be the File Set type, and therefore the set value should be specified as follows:

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

There are two types of folders: Single and File Set. If the folder type should be Single, then the single value must be specified. If the folder type should be File Set, then the set value must be specified as shown in the example.

- Specify the Folder URL using `<url>` as follows (using the proper URL):

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

- Specify the Media Name Pattern to use in the archive request using `<divaMediaNamePattern>` as follows (using the proper Media Name Pattern):

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

- Specify the Server Name using `<sourceDestinationDIVAName>` as follows (using the proper Server Name):

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

- Specify the Server Path using `<sourceDestinationDIVAPath>` as follows (using the proper Server path):

```
<sourceDestinationDIVAPath>SourceDestinationServer</sourceDestinationDIVAPath>
```

Caution: If the following Step 9 is not completed, the installation will fail because it cannot find the trace file.

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

- Install the Watch Folder Monitor service using the following command:

```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat install
```

- Start the Watch Folder Monitor service using the following command, or open the Windows Services panel and start the service from there.

```
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat start
```

Contact Telestream Support for more information and details about the full solution and configuration.

Running Multiple WFM Instances on the Same Computer

Running multiple WFM instances on the same computer requires a separate installation of DIVA for each additional WFM instance. It is possible to configure only the WFM component in these additional instances. The `<serviceName>` parameter must be

unique among all of the WFM instances. There is no special licensing required to use WFM in this configuration.

The following commands are for administration operations when running multiple WFM instances on the same computer:

Configuring the Service

The WFM configuration file is Program/conf/dfm.conf in each DIVA folder.

Upgrading the Service

Use the standard DIVA upgrade procedure to upgrade WFM.

Installing the Service

The WFM module is installed with DIVA in the same installation package.

The command line syntax to install the service using a specific configuration file is
`%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat install -conf Program/conf/dfm/dfm.conf.`

Uninstalling the Service

The command line syntax to uninstall the service using a specific configuration file is
`%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat uninstall -conf Program/conf/dfm/dfm.conf.`

Starting the Service

Execute `%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat start -conf Program/conf/dfm/dfm.conf` to start the service using a specific configuration file from the command line, or start it from the Windows Services panel.

Stopping the Service

Execute `%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat stop -conf Program/conf/dfm/dfm.conf` to stop the service running a specific configuration file from the command line, or stop it from the Windows Services panel.

Service Logging

The log folder is `%DIVA_HOME%\Program\log\dfm\`.

The logging configuration is `%DIVA_HOME%\Program\conf\dfm\dfm.trace`.

Monitoring the Service

Use the Watch Folder Monitor logs located in
%DIVA_HOME%\Program\InterLink\log\dfm\, to obtain the current status.

Additional Configuration Options

This chapter describes additional WFM configuration options and includes the following information:

Topics

- [Additional WFM Configuration](#)
- [Additional Operations WFM Configuration](#)
- [Additional Monitored Folder Configuration](#)
- [Folder Configuration Examples](#)
- [Service Log Configuration](#)
- [Trace Log Configuration](#)
- [Advance WFM Configuration](#)

Additional WFM Configuration

The following sections describe additional options for WFM configuration beyond the basic configurations described in [Solutions and Basic Configurations](#).

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 is marked as incomplete. The default value is 86400 seconds (24 hours).

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

Processing of Incomplete Files: <incompleteProcessingStrategy>

This parameter identifies how DIVA processes incomplete files. Incomplete files are files that were placed into the watch folder without the metadata file (for File Set folders), and also files that cannot be archived after a specific number of attempts.

DIVA uses the following three parameters to identify incomplete files:

- incompleteThreshold
- maxRejectCountInitializing
- maxRejectCountProcessing

The following three options are available to use for the OPTION value in the parameter as shown:

- None
This option directs the system to do nothing with an incomplete file.
- Delete
This option directs the system to delete the incomplete file.
- Rename
This option directs the system to rename the file so it is identifiable as an incomplete file.

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

MDF Configuration Priority: <mdfConfigPriority>

Note: This parameter only applies to File Set Mode folders.

This parameter can have Primary or Secondary values, and determines how the system handles the file during the archive process. This can either be by using the metadata file parameters (Primary), or the folder specific configuration parameters (Secondary). Primary is the default value used when this parameter is empty or missing.

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

Additional Operations WFM Configuration

The following sections describe additional WFM configuration options related to DIVA operations.

Delete Before Archive: <deleteBeforeArchive>

Caution: This optional parameter may lead to content deletion in the DIVA system, and must be used with caution.

This optional parameter specifies whether WFM should delete the object before the archive job for a specific folder. This parameter can be either true or false. The default value is false.

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

Example:

An object is copied to the watch folder and fits into an established Storage Plan. The Storage Plan specifies that the object should be copied to another server, and archived to a specific folder. Using this option, the object will be copied to the specified server, but then deleted before the Archive action is performed. Therefore, the object is never actually archived.

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, WFM sends the archive job to DIVA using the asterisk wildcard as the File List parameter, instead of the actual list of files for the entire folder. The value must be a positive integer to specify the number of files, or if the asterisk wildcard is used in the job, all files will be archived.

```
<numberFilesToArchiveWildcard>*</numberFilesToArchiveWildcard>
```

Object Name Conversion Rules: <objectNameConvertRules>

Note: This parameter only applies to Single File Mode folders.

The Object Name Conversion Rules identify how DIVA translates file names into object names. Currently, only the Simple conversion method is implemented. The rule syntax for this method is case sensitive and described later.

When specifying a rule using Name.Ext, the object name will be the same as the file name. If Name is only specified, the file name will be the same as the original file name, but the extension will be removed. This applies to all of the following parameters:

- `Name.Ext`
This value results in the original file name and extension; no conversion of the file name or extension takes place.
- `name` (all lowercase)
This value results in the original file name being converted to all lowercase, and the file extension is removed.
- `NAME` (all upper case)
This value results in the original file name being converted to all upper case, and the file extension is removed.
- `ext` (all lowercase)
This value results in the file name being removed, the original file name being converted to the extension, and is all lowercase.
For example, `file_name.ext` will become only `.ext`.
- `EXT` (all upper case)
This value results in the file name being removed, the original file name being converted to the extension, and is all upper case.
For example, `file_name.ext` will become only `.EXT`.
- `name.Ext`
This value results in the original file name being converted to all lowercase, and the original file extension is left intact.
- `name.EXT`
This value results in the original file name being converted to all lowercase, and the original file extension is converted to all upper case.
- `pre_name_suf.EXT`
This value results in the original file name being converted to `pre_(original_file_name)_suf`, and the original extension is converted to all upper case.

- `pre_Name_suf`

This value results in the original file name being converted to `pre_(original_file_name)_suf`, and the extension is removed.

For example, the following statement uses the Simple method, and leaves the file name and extension unchanged:

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

Original Server: <originalServer>

In some configurations, the monitored folder is an intermediate location between the video server and the DIVA system. The `<originalServer>` parameter can be specified to ensure the original content location is set correctly in the DIVA system. The original server is displayed in the GUI, and stored in the database as the original content location, instead of the actual server used to archive data from the intermediate location.

In the statement `<originalServer>original_servername</originalServer>`, the value of `original_server` is the original content location designated by the original host name. For example, the statement `<originalServer>VIDEO_SERVER_1</originalServer>` identifies `VIDEO_SERVER_1` as the original content location.

Original Path: <originalPath>

In some configurations the monitored folder is an intermediate location between the video server and the DIVA system. Specify the `<originalPath>` parameter to ensure the content path at the original location is set correctly in the DIVA system. The original content path is displayed in the GUI, and 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.

In the statement `<originalPath>original_path</originalPath>`, the value of `original_path` identifies the content path at the `original_server` location (see the previous section describing `original_server`).

Media Name: <divaMediaName> and <divaMediaNamePattern>

The `<divaMediaNamePattern>` is the Media Name Pattern used to select the medium for archive jobs. Using this pattern allows the GROUP value to be identified.

In the statement `<divaMediaNamePattern>folder1/$GROUP/folder3</divaMediaNamePattern>`, `folder1` and `folder3` are actual folder names, while `$GROUP` indicates the system will identify the Tape Group from the `.mdf` file.

If the pattern was not recognized, then the `<divaMediaName>` value from the file set's corresponding `.mdf` file is used as the Media Name for the archived job when working

with a File Set Mode folder. If the pattern is recognized, then the `<divaMediaName>` from the configuration file is used.

Additional Monitored Folder Configuration

The following sections describe additional configuration options related to the WFM monitored folder.

MDF File Extension: `<mdfExtension>`

This option identifies the extension for the metadata file for each folder. In the following example, the metadata file extension will be mdf:

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

Server Path: `<sourceDestinationDIVAPath>`

Note: This parameter only applies to File Set Watch Folders. Because the Actor and WFM could be on different machines, they might view the folder differently. For example, Actor may see the folder as a local disk, while WFM may see it as an FTP folder (or vice versa).

This field defines the location of the Watch Folder Root on the Server. For a dedicated Server (the Root Path of the server in the DIVA web app points to the root of WFM), the File Path Root of the server may already contain a path, and then leave this field empty.

In the statement `<sourceDestinationDIVAPath>path</sourceDestinationDIVAPath>`, the value of `path` is the full path to the root Server folder.

For example, the statement `<sourceDestinationDIVAPath>WATCH_FOLDER</sourceDestinationDIVAPath>` identifies `WATCH_FOLDER` as the full path to the root server folder.

Sample Server Paths for Different Configurations

In File Set Mode, the `<SourceDestinationDivaPath>` is added to the relative file path. The following examples are for different common configurations.

See [File Root Path and Single File Mode Advanced Configuration](#) for more details and examples.

Example 1:

- `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`

- Results
 - 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

Example 2:

- 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
- Results
 - 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

Example 3:

- SourceDestinationDivaPath = empty (no value entered)
- Archive File Path Root = data\tape1\123456
- Archive File List = toto.mdf toto0.data folder1\toto1.data folder1\folder2\toto2.data
- Results
 - data\tape1\123456\toto.mdf
 - data\tape1\123456\toto0.data
 - data\tape1\123456\folder1\toto1.data
 - data\tape1\123456\folder1\folder2\toto2.data

Example 4:

- SourceDestinationDivaPath = data
- Archive File Path Root = data\data\tape1\123456
- Archive File List = toto.mdf toto0.data folder1\toto1.data folder1\folder2\toto2.data
- Results
 - 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

Archive File Path Template: <archiveFilePathTemplate>

Note: This parameter is only applicable to Single File Mode Watch Folders.

This parameter specifies the rule for how the Root Path of the archive job will be generated. The platform parameter value can be one of the following:

- WIN
- SOL
- CIFS
- DETECT

The format of the value is a combination of folder names, separators, and the following keywords:

- PARENTn

This keyword identifies the folder name where the definition is located.

For example:

```
DFM_FILE_FULL_PATH = DFM_URL\PARENTn
...\PARENT_3\PARENT_2\PARENT_1\file_name.ext
```

- URL_TO_PARENTm

This keyword identifies the relative path from DFM_URL to PARENTm.

- URL_TO_FILE

This keyword identifies the relative path from DFM_URL to the file.

The following is an example using this parameter:

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

See [File Root Path and Single File Mode Advanced Configuration](#) for more details.

Archive File Name Template: <archiveFileNameTemplate>

Note: This parameter is only applicable to Single File Mode Watch Folders.

This parameter specifies the rule for identifying how the file names of an Archive job are generated. The platform parameter value can be one of the following:

- WIN
- SOL
- CIFS
- DETECT

The format of the value is a combination of folder names, separators, and the following keywords:

- PARENTn

This keyword identifies the folder name where the definition is located. For example:

```
DFM_FILE_FULL_PATH = DFM_URL\PARENTn
...\PARENT_3\PARENT_2\PARENT_1\filename.ext
```

- URL_TO_FILE

This keyword identifies the relative path from DFM_URL to the file.

- filename.ext

This keyword identifies the original file name.

- filename

This keyword identifies the file name without the extension.

In the following example filename and ext represent the keywords of the pattern definition, not the sample file name and extension; filename.ext is not an actual file name.

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

See [File Root Path and Single File Mode Advanced Configuration](#) for more details and examples.

File Filter: <fileFilter>

Note: The <fileFilter> parameter is optional for the watch folder configuration. If it is not specified in the folder configuration, WFM will archive any file in the folder without using any masks. Only one <fileFilter> can be specified per folder.

Use this parameter to filter files to archive. The type attribute can be specified as either include or exclude.

In inclusion mode, no files are 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 WFM, and not deleted or archived.

In exclusion mode, all files are 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 WFM, and not deleted or archived.

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

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

- <mask>
This parameter specifies the pattern for the filter.
- <regexp>
This parameter specifies the regular expression for the filter.
For details on Regular Expressions visit the following websites:
https://en.wikipedia.org/wiki/Regular_expression
<http://www.regular-expressions.info/>

Delete Parent and Content Directories: <deleteParentDirectoryAndContentDirectories>

This optional parameter specifies whether WFM should delete the parent and child folders of an MDF file. This is useful when a user places an object into a folder, and WFM is not set to delete the files and folders after all actions are executed on the object. Using this parameter, the object's files and folders are not left in the folder where the object was placed. This parameter can be either TRUE (this is the default) or FALSE.

In the following example, the folders will not be deleted:

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

In the following example, the folders will be deleted:

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

Folder Configuration Examples

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

Example of a Local Folder in File Set Mode

The Object Name Conversion Rule does not apply to Set Mode Folders. Files will be acted on using the original file name and extension.

In the following configuration example:

- The local folder is C:\WATCHFOLDER.
- The type of WFM folder is File Set.
- The Priority is 50.
- The Collection is Collection1.
- The Incomplete Threshold is 60 seconds.
- The Server name is LOCAL1.
- The Server path is C:/WATCHFOLDER.
- The Media Name is Array1.
- The Incomplete File Processing Strategy is Delete.
All incomplete folders will be deleted.

```
<folderConfig>  
<url>file:///c:\WATCHFOLDER</url>  
<type>set</type>  
<priority>50</priority>  
<CategoryName>Collection1</CategoryName>  
<incompleteThreshold>60</incompleteThreshold>  
<sourceDestinationDIVAName>LOCAL1</sourceDestinationDIVAName>  
<sourceDestinationDIVAPath>C:/WATCHFOLDER</  
sourceDestinationDIVAPath>  
<divaMediaName>Array1</divaMediaName>  
<incompleteProcessingStrategy>Delete</  
incompleteProcessingStrategy>  
</folderConfig>
```

Using this configuration, if a set of files is copied into the WFM monitored local folder named C:\WATCHFOLDER, and it has 3 files in the set with the names file1.ext, file2.ext, and filesetdata.mdf, they will be archived into a single file in DIVA with the following parameters:

- The Server is LOCAL1.
- The Object Name is filesetdata.ext.
- The Collection Name is Collection1.

If the metadata file is not present, the set will be considered incomplete and the files will all be deleted after 60 seconds.

Example of an FTP Folder in Single File Mode

When the WFM watch folder is configured on an FTP server, all content expected to be archived must be copied to the watch folder using FTP. If the content is copied to the watch folder by other means (local copy through the operating system, remote desktop copy, and so on) there is no guarantee of proper processing by WFM.

When WFM 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 Watch Folder Monitor Configuration: ftp://diva:password@host_ip/folder1
- Incorrect Watch Folder Monitor Configuration: ftp://diva:password@host_ip/ifs/folder1

```
<folderConfig>
<url>ftp://diva:diva@localhost:21/dropfolder</url>
<type>single</type>
<priority>20</priority>
<CategoryName>Collection2</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 this example:

- The FTP folder is ftp://diva:diva@localhost:21/watchfolder.
- The Type of WFM folder is Single.
- The Priority is 20.
- The Collection is Collection2.
- The Incomplete Threshold is 120 seconds.
- The Server Name is FTP1.
- The Server Path is /video1/data/.
- The Media Name is Array1.
- The default Object Name Conversion rule for watch folders will remove the file extension.
- The divaMediaNamePattern is /def/\$GROUP/content.

Using this configuration, if a single file is transferred to the FTP server called localhost, on port 21, with the login name diva, the password diva, in the WFM-monitored FTP

folder named `watchfolder`, with the file name `file1.ext`, it will be archived to a single object with the following parameters:

- Server: FTP1
- Object Name: file1
- Collection Name: Collection2

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.

Example of a CIFS Folder in File Set Mode

The following example is for when using a CIFS folder for the watch folder. The Object Name Conversion Rule does not apply to Set Mode Folders. Files will be acted on using the original file name and extension.

In the following example:

- The CIFS folder is `\\DIVAServer\watchfolder`.
- The Type of WFM folder is File Set.
- The Priority is 50.
- The Collection is Collection3.
- The Incomplete Threshold is 60 seconds.
- The Server Name is CIFS1.
- The Server Path is empty.
- The Media Name is array1.
- The incomplete file processing strategy is Rename.

All incomplete folders will be renamed to `failed_ + (original_file_name)`.

```
<folderConfig>
<url>file:///\\tscmsserver\watchfolder</url>
<type>set</type>
<priority>50</priority>
<CategoryName>Collection3</CategoryName>
<incompleteThreshold>60</incompleteThreshold>
<sourceDestinationDIVAName>CIFS1</sourceDestinationDIVAName>
<sourceDestinationDIVAPath></sourceDestinationDIVAPath>
<divaMediaName>array1</divaMediaName>
<incompleteProcessingStrategy>Rename</
  incompleteProcessingStrategy>
</folderConfig>
```

Using this configuration, if a set of files is copied into the WFM-monitored CIFS folder named `watchfolder`, on the computer named `divaserver`, and has three files in the set with the names `file1.ext`, `file2.ext`, and `filesetdata.mdf`, they will be archived to a single object with the following parameters:

- Server: CIFS1
- Object Name: filesetdata.ext

- Collection Name: Collection3

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

Service Log Configuration

Use the following parameters to specify the wrapper log file tracing configuration in Program/conf/dfm/dfm.conf.

- `wrapper.logfile.loglevel`
This parameter specifies the level of messages to be logged.
- `wrapper.logfile.maxsize`
This parameter specifies the maximum size of the log file in bytes, kilobytes, or megabytes.
- `wrapper.logfile.maxfiles`
This parameter specifies the maximum log files count.

The valid logging levels are as follows:

- NONE
This logging level produces no output.
- FATAL
This logging level produces only messages for irrecoverable errors.
- ERROR
This logging level produces all error messages.
- WARN
This logging level produces all warning messages.
- STATUS
This logging level produces all state changes.
- INFO
This logging level produces all JVM output and informative messages.
- DEBUG
This logging level produces detailed debugging information.

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
```

Trace Log Configuration

Use the following parameters to specify the tracing configuration for Watch Folder Monitor in the file Program/conf/dfm/dfm.trace:

- `LOGGING_TRACE_LEVEL`
This parameter specifies the level of messages to be logged.
- `LOGGING_MAXFILESIZE`
This parameter specifies the maximum size of the log file in bytes, kilobytes, or megabytes.
- `LOGGING_LIFETIME`
The maximum lifetime of a logged item, in hours.

The following is a sample dfm.trace configuration:

```
-----  
# Levels can be: DEBUG, INFO, WARN, ERROR, FATAL  
  
LOGGING_TRACE_LEVEL=INFO  
LOGGING_MAXFILESIZE=7MB  
LOGGING_LIFETIME=50  
-----
```

Advance WFM Configuration

Caution: The parameters in the following sections should not be altered unless required, and by somebody confident of having the ability to do so correctly. Misconfiguration of these parameters could result in an unstable, or non-functioning system, or both! Telestream recommends saving the original configuration file to a different file name before making any modifications to the configuration (for example, `wfm.conf.original.ini`).

Full WFM Configuration File

There are several groups of parameters for the configuration file as follows:

- Global WFM Parameters
These parameters specify common behavior of WFM.
- Folders Default Parameters
These parameters specify some default values for folder specific parameters.
- Folder Specific Parameters
These parameters specify folder specific parameters.
- Tracing Specific Parameters
These parameters specify tracing parameters.

See [Full Watch Folder Monitor Configuration File \(`dfm.conf.ini`\)](#) to view a sample of the full configuration file.

Global WFM Configuration

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

- `<managerConnetion>`
This parameter specifies how WFM connects to DIVA. These options are not required to be defined unless the DIVA system being connected to requires connection parameters outside of the default configuration.
For example:

```
<address host="..." port="..." userName="..." password="..." applicationName="..." />
```

 - `host`
This parameter is required and identifies the address of DIVA.
 - `port`
This parameter is required and identifies the port number where DIVA is listening. This must be a positive integer value.

- `userName`
 This parameter is optional and identifies the user name to log in to the DIVA system. If the value is NULL or empty, DIVA uses the default user name and privileges from the DIVA system for this application.
- `password`
 This parameter is optional and identifies the password for the user. The connection is rejected if the provided password is incorrect.
- `applicationName`
 This parameter is optional and identifies an arbitrary string that is interpreted by DIVA as an application name. This is used for information purposes only, and can be NULL or an empty string.
- `<serviceName>`
 This parameter identifies the WFM Service Name for the currently installed instance. Spaces are not allowed within the Service Name. The Service Name will be DIVA Watch Folder Monitor - serviceName. If serviceName is left empty, the Service Name DIVA Watch Folder Monitor is used.
- `<foibStateFile>`
 This parameter identifies the file name of the File/Object Information Base dump.
- `<doibStateFile>`
 This parameter identifies the file name of the Delete File/Object Information Base dump.
- `<fileReloadingThreshold>`
 This parameter identifies the file reloading threshold in seconds. This interval specifies how often the File Monitor module reloads the state of the WFM folders. This setting must be a positive integer value between 1 and 10,000,000.
- `<fileReadyThreshold>`
 This parameter identifies the time interval, in seconds. If the last modification time to a file remains unchanged after this interval, the file is marked as copied, and then archived. This setting must be a positive integer value between 1 and 10,000,000.
- `<dirDeleteThreshold>`
 This parameter identifies the Directory Delete Threshold in seconds. This interval specifies how often the File Monitor module reloads the state of the WFM folders in Delete Mode, and commences with deletion. This setting must be a positive integer value between 1 and 10,000,000.
- `<fileDeleteThreshold>`
 This parameter identifies the File Delete Threshold in seconds. This interval specifies when a file will be deleted after copying is complete for WFM folders in Delete Mode. This setting must be a positive integer value between 1 and 10,000,000.
- `<maxArchiveOperations>`
 This parameter identifies the maximum number of simultaneous archive operations that WFM can perform. If the number of jobs exceeds this threshold, pending

jobs must wait until all active archive operations are complete. This setting must be a positive integer value between 1 and 10,000,000.

- `<maxRejectCountInitializing>`
 This parameter identifies the maximum number of archive retries that WFM can perform for jobs rejected by DIVA during job initialization. Objects exceeding this limit are marked as incomplete. This setting must be a positive integer value between 1 and 10,000,000.
- `<maxRejectCountProcessing>`
 This parameter identifies the maximum number of archive retries that WFM can perform for jobs rejected by DIVA during job processing. Objects exceeding this limit are marked as incomplete. This setting must be a positive integer value between 1 and 10,000,000.
- `<pollingIntervalOfRequestState>`
 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.
- `<repeatRequestDIVAThreshold>`
 This parameter identifies the interval in seconds, before DIVA's Archive module will resubmit rejected jobs. This setting must be a positive integer value between 1 and 10,000,000.

Folders Default Configuration

The Folders Default configuration parameter values are used for folders in a folder specific configuration when a parameter is not specified. All default values can be overridden within the folder specific configuration. The Folders Default configuration accepts the following parameters:

- `<type>`
 There are two types of folders; Single File and File Set. To configure the folder for Single Files, specify single. To configure the folder for File Sets, specify set.
- `<mode>`
 This parameter identifies the WFM folder work mode. This can be Archive or Delete.
- `<priority>`
 This parameter identifies the default priority level of the jobs for the folder. The folder priority range is 0 - 100. The value 0 is the lowest priority, and 100 is the highest.
- `<CategoryName>`
 This parameter identifies the default value for the Collection Name. This value will be provided to DIVA if it is not overridden in the folder specific configuration, or in the File Set metadata file.

- `<incompleteThreshold>`
 This parameter identifies the Incomplete Threshold in seconds. It specifies the maximum amount of time that a Single File, or File Set, can reside in the watch folder and not be archived before it is marked as incomplete. This must be a positive integer value between 1 and 10,000,000.
- `<sourceDestinationDIVAName>`
 This parameter identifies the Server Name to use in the Archive job.
- `<sourceDestinationDIVAPath>`
 This parameter identifies the Server Path to use in the Archive job.
- `<mdfExtension>`
 This parameter identifies the extension of the metadata file for each folder.
- `<originalServer>`
 This parameter identifies the server to store in the object catalog if different from the actual server. This is useful when using an intermediate storage area for archiving, but storing the original source for the data in the object catalog information.
- `<originalPath>`
 This parameter identifies the File Root Path to store in the object catalog, if different from the actual File Root Path. This is useful when using an intermediate storage area for archiving, but storing the original source for the data in the object catalog information.
- `<incompleteProcessingStrategy>`
 This parameter identifies the incomplete folder and file processing strategy. The value can be None, Delete, or Rename.
- `<numberFilesToArchiveWildcard>`
 This parameter identifies the threshold for the number of files in the same folder. If the number of files in the folder becomes larger than the `<numberFilesToArchiveWildcard>` value, WFM sends the Archive job to DIVA using the asterisk wildcard for the entire folder. This value must be a positive integer.
- `<divaMediaNamePattern>`
 The Media Name Pattern specifies the archive medium by extracting a part of the archive Root Path. This option is mutually exclusive with the `<divaMediaName>` option.
- `<deleteBeforeArchive>`
 This parameter specifies whether WFM should delete the object from DIVA before the archive operation, if the object already exists. This parameter can be set to either TRUE or FALSE. This is an optional parameter and the default value is FALSE.
- `<objectNameConvertRule method="simple">Name.Ext</objectNameConvertRule>`
 The value of `<objectNameConvertRule>` specifies the rule for the algorithm specified. In the current version of WFM, only the Simple conversion method is

implemented. See the [Object Name Conversion Rules: <objectNameConvertRules>](#) section for the rule syntax for this method.

Folder Specific Configuration

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

- `<url>`
This parameter identifies the WFM folder URL. This is the monitored folder as viewed by the WFM process.
- `<type>`
There are two types of folders; Single File and File Set. To configure the folder for Single Files, specify single. To configure the folder for File Sets, specify set.
- `<mode>`
This parameter identifies the WFM folder work mode. This can be Archive or Delete.
- `<priority>`
This parameter identifies the default priority level of the jobs for the folder. The folder priority range is 0 - 100. The value 0 is the lowest priority, and 100 is the highest.
- `<CategoryName>`
This parameter identifies the default value for the Collection Name. This value will be provided to DIVA if it is not overridden in the File Set metadata file.
- `<incompleteThreshold>`
This parameter identifies the Incomplete Threshold in seconds. This parameter specifies the maximum amount of time that a Single File, or File Set, can reside in the watch folder and not be archived before it is marked as incomplete. This must be a positive integer value between 1 and 10,000,000.
- `<sourceDestinationDIVAName>`
This parameter identifies the Server Name to use in the Archive job.
- `<sourceDestinationDIVAPath>`
This parameter identifies the Server Path to use in the Archive job.
- `<divaMediaName>`
This parameter identifies the Media Name to use in the Archive job. This option is mutually exclusive with the `<divaMediaNamePattern>` option.
- `<mdfExtension>`
This parameter identifies the extension of the metadata file for each folder.
- `<originalServer>`
This parameter identifies the server to store in the object catalog if different from the actual server. This is useful when using an intermediate storage area for

archiving, but storing the original source for the data in the object catalog information.

- `<originalPath>`
This parameter identifies the File Root Path to store in the object catalog, if different from the actual File Root Path. This is useful when using an intermediate storage area for archiving, but storing the original source for the data in the object catalog information.
- `<incompleteProcessingStrategy>`
This parameter identifies the incomplete folder and file processing strategy. The value can be None, Delete, or Rename.
- `<numberFilesToArchiveWildcard>`
This parameter identifies the threshold for the number of files in the same folder. If the number of files in the folder becomes larger than the [Number of Files to Archive Wildcard: `<numberFilesToArchiveWildcard>`](#) value, WFM sends the Archive job to DIVA using the asterisk wildcard for the entire folder. This value must be a positive integer.
- `<divaMediaNamePattern>`
The Media Name Pattern specifies the archive medium by extracting a part of the archive Root Path. This option is mutually exclusive with the `<divaMediaName>` option.
- `<deleteBeforeArchive>`
This parameter specifies whether WFM should delete the object from DIVA before the archive operation, if the object already exists. This parameter can be set to either TRUE or FALSE. This is an optional parameter and the default value is FALSE.
- `<objectNameConvertRule method="simple">Name.Ext</objectNameConvertRule>`
The value of `<objectNameConvertRule>` specifies the rule for the algorithm specified. In the current version of WFM, only the Simple conversion method is implemented. See the [Object Name Conversion Rules: `<objectNameConvertRules>`](#) section for the rule syntax for this method.

File Root Path and Single File Mode Advanced Configuration

The WFM configuration receives file information from the remote or local media (FTP server, shared directory, mounted drive, and so on). Based on the file name and folder, WFM generates the File Root Path and file name.

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

Because requirements for the file name vary, the system can either include or exclude, the parent folder in the file. The requirements for the path also vary because there are several ways the Actor can connect to the media.

The File Root Path must be specified in Single File Mode in the following format:

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

The file name must be specified in Single File Mode in the following format:

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

WFM monitors a Watch Folder Monitor URL (FTP, CIFS, Windows disk, or Solaris disk). WFM obtains a File Full Path (for FTP, CIFS, Windows disk, or Solaris disk) for each incoming file.

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

or

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

Note: The operating system determines whether to use / or \ for these statements.

```
<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_PARENTm.

```
<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_PARENTm, URL_TO_FILE, Filename.Ext, and Filename, where the platform is WIN, SOL, CIFS, or DETECT.

Available keywords are as follows:

- PARENTn
 This keyword identifies the folder name where the definition is located. For example:

```
WFM_FILE_FULL_PATH = WFM_URL\PARENTn
...\PARENT_3\PARENT_2\PARENT_1\Filename.ext
```
- URL_TO_PARENTm
 This keyword identifies the path from WFM_URL to PARENTm.
- URL_TO_FILE
 This keyword identifies the relative path from WFM_URL to the file.
- Filename.ext
 This keyword identifies the original file name.

- `Filename`

This keyword identifies the file name without the extension.

It is important to use the correct case in all of the following examples because the file name and extension parameters are case sensitive. For example, `filename.ext` is different than `Filename.Ext`.

Workflow A

The files are located in the root of `WFM_URL`.

- The watch folder is on CIFS
- The Actor is on Windows

`WFM_URL = file:///\\aaa\bbb\ccc\ddd`

`File Full Path = \\aaa\bbb\ccc\ddd\ffffff.txt`

- The Actor sees the Watch Folder Monitor 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.

If some files are located in the root of the `WFM_URL`, and some are located in subfolders, the path length is not constant.

This can be addressed by using `ARCHIVE_FILE_NAME_TEMPLATE = "URL_TO_FILE\Filename.Ext"`. When the `URL_TO_FILE` parameter is empty, WFM automatically removes the backslash character. An empty `ARCHIVE_FILE_PATH_TEMPLATE` value is supported.

Workflow B

If Watch Folder Monitor monitors the video server using FTP, the file must have a file extension, and the Actor will archive through the server interface without a file extension.

- The file must be located in a single folder level
- The watch folder is on FTP

When the WFM watch folder is configured on an FTP server, all content expected to be archived must be copied to the watch folder using FTP. If the content is copied to the watch folder by other means (local copy through the operating system, remote desktop copy, and so on) there is no guarantee of proper processing by Watch Folder Monitor.

- The Actor is using a special interface

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

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

- The Actor sees the Watch Folder Monitor 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.Ext"
```

- This is how the file extension is removed.

Administration, Operations and Monitoring

This chapter describes Watch Folder Monitor administration, operations, and monitoring, and includes the following information:

Topics

- [Starting, Stopping, and Restarting Watch Folder Monitor Overview](#)
- [Summary of WFM Administration](#)
- [Command-Line Interface](#)
- [Operations and Workflows](#)
- [Monitoring Watch Folder Monitor](#)

Starting, Stopping, and Restarting Watch Folder Monitor Overview

The WFM service can be started, stopped and restarted using the operating system services facilities, or from the WFM command line utility as follows:

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

The command line options are as follows:

- `start`
Tells the Watch Folder Monitor service to start
- `stop`
Tells the Watch Folder Monitor service to stop
- `restart`
Tells the Watch Folder Monitor service to stop, and then start again

Starting and Restarting the Watch Folder Monitor Service

When the Watch Folder Monitor service starts, or restarts, WFM loads and validates the configuration file. If WFM detects any configuration issues, the process terminates and runs a diagnostics routine.

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

If the configuration validation is successful, WFM scans all of the configured watch folders, then checks the status of all objects initialized before WFM was last shutdown, and then updates the internal database with the current status of the objects.

After all of these steps have completed, the WFM status will be "Running".

Running the Watch Folder Monitor Service

Caution: The last file to be created in the File Set Watch 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!

When WFM locates files in a configured watch folder, it updates the internal database and requests DIVA to archive all files found as new objects. In order to avoid repeated archive jobs, WFM continuously updates the archive operations status in the internal database.

WFM logs information about the incomplete files and calls the WFM File DIVA module to move them to the trash folder.

If DIVA jobs fail, the DIVA Status module informs the internal database of the failure. If the number of unsuccessful job attempts reaches a preconfigured number, the object status is changed to “Could not be archived” and the object is marked as incomplete.

If the job completes successfully, the internal database is updated by the DIVA Status module. WFM removes the MDF file, and the File Set folder, for a File Set object.

Stopping the Watch Folder Monitor Service

Executing the shutdown script will terminate the WFM service. All internal processes are stopped before all archive operations are completed. After all of the modules are stopped, all internal statuses are saved to disk in the internal database before WFM completes shut down.

Summary of WFM Administration

The following is an overview of the administrative functions available in WFM. There is no special licensing required to use WFM.

Configuring the Service

The WFM configuration file is %DIVA_HOME%\Program\conf\dfm\dfm.conf.

Upgrading the Service

Use the standard DIVA upgrade procedure to upgrade WFM.

Installing the Service

The WFM module is installed with DIVA in the same installation package.

The command line syntax to install the service is
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat install.

Uninstalling the Service

The command line syntax to uninstall the service is
%DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat uninstall.

Starting the Service

Execute %DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat start to start the service from the command line, or start it from the Windows Services panel.

Stopping the Service

Execute %DIVA_HOME%\Program\InterLink\dfm\bin\dfm.bat stop to stop the service from the command line, or stop it from the Windows Services panel.

Service Logging

The log folder is %DIVA_HOME%\Program\log\dfm\.

The logging configuration is %DIVA_HOME%\Program\conf\dfm\dfm.trace.

Monitoring the Service

Use the WFM logs, located in %DIVA_HOME%\Program\InterLink\log\dfm\, to obtain the current status.

Command-Line Interface

Commands can be initiated using the operating system command-line. The syntax and options are as follows:

```
dfm {install|uninstall|start|stop|restart|status|version|help} [-conf|-f]
```

The commands are as follows:

- `install`
This command installs the module as a system service.
- `uninstall`
This command removes the executable as a system service.
- `start`
This command starts the service.
- `stop`
This command stops the service if it is currently running.
- `restart`
This command stops, and then subsequently starts the service.
- `status`
This command displays the current status of the service.
- `version`
This command displays the module release information, and then exits.
- `help`
This command displays the help information, and then exits.

The options are as follows:

- `-conf`
This option identifies the configuration file to load the Watch Folder Monitor settings from.

- `-f`

This option is an alternative to using the `-conf` option and performs the same function.

Operations and Workflows

Watch Folder Monitor constantly monitors the configured watch folders. When a file, or file set, is found in one of the configured watch folders, it is entered into the internal database (the internal database is an XML file) with the file size, date, and time when the file was originally found. After waiting for a configured period, WFM 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 watch folder configured in Single File Mode satisfies the applicable criteria in the WFM folder configuration (that is, it was not modified within the previous 60 seconds), WFM marks the file as complete and sends an Archive job to the DIVA system. If the file has not fulfilled the criteria for a complete file after a configured allowable period, then it is marked as incomplete and set for deletion.

Caution: The last file to be created in the File Set Watch 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!

In File Set Folders, after the metadata file (.mdf) is located in the watch folder, WFM marks all of the files in the File Set as complete, and initiates the Archive job to DIVA. If the metadata file does not exist after a configured period, then all files are marked incomplete and set for deletion.

By default, WFM will try to archive any files located in the configured and monitored watch folders. If the files cannot be archived according to the WFM Folder configuration, they are 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 interval for how often the watch 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, archived, or both, within the specified time period are flagged as an unsuccessful archive, an incomplete object, or deleted.

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

Source server files cannot be deleted from a watch folder after an archive completes unless the appropriate Server is configured using the `-allow_delete_on_source` option.

See [Additional Configuration Options](#) for details on configuration options, features, and examples.

Algorithm and Workflow for Single File Mode Watch Folders

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

To begin the process, WFM checks if the file size is equal to zero, 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.

Next, WFM 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` value used in this check is the local WFM time when the last modification of size, or date, was captured by WFM.

WFM checks if the difference between the current time (local WFM time) and the `lastModificationTime` is greater than the `fileReadyThreshold` parameter setting in the configuration file, whether the file is locked for processing, and whether the disk has free space available.

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

Note: An unknown status indicates that all required file information was not added to WFM. WFM 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 WFM. It will only become known after all file information has been added to WFM for processing.

Example 1

The following is an example workflow:

1. The file `test.mpg` was found by WFM, and the date and size was stored.
2. The file `test.mpg` was found by WFM again (during the next cycle of folders investigation), the date and size was changed from the previous check. Therefore, the file will not be added as known to WFM.
3. The file `test.mpg` was found by WFM again (during the next cycle of folders investigation), the date and size was not changed from the previous check, the difference is greater than the `fileReadyThreshold` parameter setting, the file is unknown to WFM, not locked for processing, and disk space is available. Therefore, the file was added as known to WFM.

4. The Archive job was sent to DIVA.
5. The archive was complete.

Example 2

The following is an example workflow:

1. The file test.mpg was found by WFM, and the date and size was stored.
2. The file test.mpg was found by WFM again (during the next cycle of folders investigation), the date and size was changed from last check. Therefore, the file will not be added as known to WFM.
3. The file test.mpg was found by WFM again (during the next cycle of folders investigation), the date and size was not changed from the previous check, the difference is greater than the `fileReadyThreshold` parameter setting, the file is unknown to WFM, not locked for processing, and disk space is available. Therefore, the file was added as known to WFM.
4. The Archive job was sent to DIVA.
5. The archive was not completed. Therefore, it was terminated by DIVA for some reason.
6. DIVA attempts to archive again after `<repeatRequestDIVAThreshold>` seconds, but the job was rejected again.
7. If the difference is greater than the `fileReadyThreshold` parameter setting, WFM knows about the file for longer than the `incompleteThreshold` parameter setting, and it is not locked for processing, then WFM will mark the file as incomplete. Otherwise, the system will continue to retry starting with Step 4.

Algorithm and Workflow for File Set Mode Watch Folders

For File Set Mode folders, Watch Folder Monitor 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, WFM first checks to find out if the file size is equal to zero 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.

WFM 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 Watch Folder Monitor time when the last modification of size or date was captured by WFM.

WFM checks if the difference between the current time (local Watch Folder Monitor time) and the `lastModificationTime` is greater than the `fileReadyThreshold` parameter setting in the configuration file, whether the file is locked for processing, and if the disk has free space available.

If the difference is greater than the `fileReadyThreshold` parameter setting, the file is unknown to WFM and not locked for processing, then WFM directs DIVA to archive the file.

If the difference is greater than the `fileReadyThreshold` parameter setting, the file is known to WFM, not locked for processing, the file archived successfully, and the difference between the current time (local Watch Folder Monitor time) and the archive time is greater than 5 seconds, or the file should be deleted as incomplete, then WFM will delete the file.

If the difference is greater than the `fileReadyThreshold` parameter setting, WFM knows about the file for longer than the `incompleteThreshold` parameter setting, and it is not locked for processing, then WFM will mark the file as incomplete.

Note: An “unknown” status indicates that all required file information was not added to WFM. WFM 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 WFM. It will only become known after all file information has been added to WFM for processing.

Example 1

The following is an example workflow:

1. The file test.mdf was found by WFM, and the date and size was stored.
2. The file test.mdf was found by WFM again (during the next cycle of folders investigation), and the date and size was changed from the previous check. Therefore, the file will not be added as known to Watch Folder Monitor.
3. The file test.mdf was found by WFM again (during the next cycle of folders investigation), the date and size was not changed from the previous check, the difference is greater than the `fileReadyThreshold` parameter setting, the file was unknown to WFM. Therefore, the file was added as known to Watch Folder Monitor.
4. The Archive job is sent to DIVA.
5. The archive was completed.
6. The file test.mdf was found by WFM again (during the next cycle of folders investigation), the date and size was not changed from the previous check, the difference is greater than the `fileReadyThreshold` parameter setting, the file is known to WFM, not locked for processing, the file archived successfully, and the difference between the current time (local Watch Folder Monitor time) and the archive time is greater than 5000. Therefore, the file was deleted by Watch Folder Monitor.

Example 2

The following is an example workflow:

1. The file test.mdf was found by WFM, and the date and size was stored.
2. The file test.mdf was found by WFM again (during the next cycle of folders investigation), the date and size was changed from the previous check. Therefore, the file will not be added as known to Watch Folder Monitor.

3. The file test.mdf was found by WFM again (during the next cycle of folders investigation), the date and size was not changed from the previous check, the difference is greater than the `fileReadyThreshold` parameter setting, the file is unknown to WFM. Therefore, the file was added as known to Watch Folder Monitor.
4. The Archive job was sent to DIVA.
5. The archive was not completed, and terminated by DIVA for some reason.
6. DIVA attempts to archive again after `<repeatRequestDIVAThreshold>` seconds, but the job was rejected again.
7. If the difference is greater than the `fileReadyThreshold` parameter setting, WFM knows about the file for longer than the `incompleteThreshold` parameter setting, and it is not locked for processing, then WFM will mark the file as incomplete. Otherwise the system will continue to retry starting with Step 4.

Monitoring Watch Folder Monitor

The following sections describe different methods for monitoring the Watch Folder Monitor service.

Monitoring DIVA Connect from the Web App

You use the DIVA web app to monitor jobs initiated by the Watch Folder Monitor. Open the web app in Chrome at http://<system_mgmt_app_ip>:8765/.

The DIVA web app enables viewing the status of the jobs, job history, and other portions of the system. The Jobs, Job History, Catalog Browsing, and SPM Actions pages under the Content Management menu offer filtering functionality. The Content Management > Catalog Browsing page displays the information in the table columns. The table columns being viewed are editable using the Edit Table icon on the top right of the page.

Use the web app Job History page to query completed WFM archive jobs. Searches can be filtered using the filtering icon on the top of the page. To make querying WFM archive jobs easier, assign a dedicated server exclusively for Watch Folder Monitor (possibly with the same parameters as another server in the system). However, note this technique will not work for DIVA delete jobs issued by WFM.

Monitoring WFM using Logs

The Watch Folder Monitor log files must be used for status monitoring of WFM. The files are located in the `%DIVA_HOME%\Program\InterLink\log\dfm\` folder. The log files contain detailed information about files and folders encountered by WFM, information about jobs submitted to DIVA, files/folders marked as incomplete, and errors encountered in processing.

See [Sample of the Full Trace Log File](#) for detailed information about this file.

Configuration Files and Examples

This appendix includes default configuration files and configuration examples, and includes the following information:

Topics

- [Configuring Objects](#)
- [Metadata File](#)
- [Sample of the Full Trace Log File](#)
- [Full Watch Folder Monitor Configuration File \(dfm.conf.ini\)](#)

Configuring Objects

All File Set Objects are described by a companion MDF (Metadata File). Object configuration must be specified in the MDF file using the following parameters:

- `priority`
This parameter identifies the priority of the DIVA job 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`
This parameter identifies the Object Collection Name. Watch Folder Monitor provides this value to DIVA.
- `objectName`
This parameter identifies the Object Name. Watch Folder Monitor provides this value to DIVA.
- `comments`
This parameter identifies optional information describing the object. Watch Folder Monitor provides this value to DIVA.
- `sourceDestinationDIVAName`
This parameter identifies the Server Name used in the archived job.
- `sourceDestinationDIVAPath`
This parameter identifies the Server Path used in the archived job.
- `originalServer`
This parameter identifies the server to store in the object catalog, if different from the actual server. This is useful when using an intermediate storage area for archiving, but storing the original source for the data in the object catalog information.
- `originalPath`
This parameter identifies the File Root Path to store in the object catalog, if different from the actual File Root Path. This is useful when using an intermediate storage area for archiving, but storing the original source for the data in the object catalog information.
- `fileList`
This parameter identifies the list of File Path Names that should be archived relative to the folder specified in `sourceDestinationDIVAPath` parameter.

Metadata File

Caution: The last file to be created in the File Set Watch 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!

Create the MDF (Metadata File) in the original MDF format, or in XML format. Initially, WFM will attempt to parse the MDF as an XML file. If it fails, then it will parse it in the original MDF format.

Sample of the sample.mdf File in XML Format

The following is an example of the MDF file in the XML format:

```
#
#Object configuration.
#
<DIVAObjectDefinition>
  <priority>50</priority>
  <CategoryName>Collection1</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>
```

Sample of the sample.mdf File in Original MDF Format

The following is an example of the MDF file in the original format:

```
#
#Object configuration.
#
priority=50
CategoryName=Collection1
objectName=object name
<comments>
  here are comment lines
</comments>
```

```
sourceDestinationDIVAName=SD1
sourceDestinationDIVAPath=/video1/data/

originalServer=original server
originalPath=original path

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

Sample of the Full Trace Log File

The following is an example of the full Trace log file:

```
08/02 15:53:12.263 INFO [main,] TraceCore: Logging (re)configured
with: file:/E:/app/DIVA/7.7.22.0/Program/InterLink/wfm/bin/../../
../conf/wfm/wfm.trace, found as: file specified in
setConfigurationPath()
=====
====
08/02 15:53:12.283 INFO [main,] TraceCore: Logging (re)configured
with: file:/E:/app/DIVA/7.7.22.0/Program/InterLink/wfm/bin/../../
../conf/wfm/wfm.trace, found as: file specified in
setConfigurationPath()
=====
====
08/02 15:53:12.465 INFO [Wrapper-Connection,] TraceCore: Logging
(re)configured with: file:/E:/app/DIVA/7.7.22.0/Program/InterLink/
wfm/bin/../../../conf/wfm/wfm.trace, found as: file specified in
setConfigurationPath()
=====
====
08/02 15:53:12.482 INFO [Wrapper-Connection,] TraceCore: Logging
(re)configured with: file:/E:/app/DIVA/7.7.22.0/Program/InterLink/
wfm/bin/../../../conf/wfm/wfm.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,] TraceCore: Logging
(re)configured with: file:/E:/app/DIVA/7.7.22.0/Program/InterLink/
wfm/bin/../../../conf/wfm/wfm.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: Watch 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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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 prépa
meeting.docx from folder ftp://anonymous:anonymous@127.0.0.1/ is
accepted!
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
WFM...
08/02 16:05:34.408 INFO [Wrapper-Connection,] TraceCore: Logging
(re)configured with: file:/E:/app/DIVA/7.7.22.0/Program/InterLink/
wfm/bin/../../../../conf/wfm/wfm.trace, found as: file specified in
setConfigurationPath()
=====
08/02 16:05:34.410 INFO [Wrapper-Connection,] Stop: Stopping Watch
Folder Monitor...
08/02 16:05:34.410 DEBUG [Wrapper-Connection,] WFMModuleThread:
Stopping module class
com.storagetek.diva.proxy.wfm.FileMonitorThread...
08/02 16:05:34.410 DEBUG [Wrapper-Connection,] WFMModuleThread:
Module class com.storagetek.diva.proxy.wfm.FileMonitorThread
stopped.
08/02 16:05:34.410 DEBUG [Wrapper-Connection,] WFMModuleThread:
Stopping module class
com.storagetek.diva.proxy.wfm.DMFileMonitorThread...
08/02 16:05:34.410 DEBUG [Wrapper-Connection,] WFMModuleThread:
Module class com.storagetek.diva.proxy.wfm.WMFileMonitorThread
stopped.
08/02 16:05:34.410 DEBUG [Wrapper-Connection,] WFMModuleThread:
Stopping module class
com.storagetek.diva.proxy.wfm.DIVAArchiveThread...
08/02 16:05:34.410 DEBUG [Wrapper-Connection,] WFMModuleThread:
Module class com.storagetek.diva.proxy.wfm.DIVAArchiveThread
stopped.
08/02 16:05:34.410 DEBUG [Wrapper-Connection,] WFMModuleThread:
Stopping module class
com.storagetek.diva.proxy.wfm.DIVAStatusThread...
08/02 16:05:34.411 DEBUG [Wrapper-Connection,] WFMModuleThread:
Module class com.storagetek.diva.proxy.wfm.DIVAStatusThread
stopped.
08/02 16:05:34.411 DEBUG [Wrapper-Connection,] WFMModuleThread:
Stopping module class
com.storagetek.diva.proxy.wfm.IncompleteLogThread...
08/02 16:05:34.411 DEBUG [Wrapper-Connection,] WFMModuleThread:
Module class com.storagetek.diva.proxy.wfm.IncompleteLogThread
stopped.
08/02 16:05:34.411 DEBUG [Wrapper-Connection,] WFMModuleThread:
Stopping module class
com.storagetek.diva.proxy.wfm.common.config.WFMConfigUpdater...
08/02 16:05:34.411 DEBUG [Wrapper-Connection,] WFMModuleThread:
Module class
    
```

```

com.storagetek.diva.proxy.wfm.common.config.WFMConfigUpdater
stopped.
08/02 16:05:34.411 DEBUG [Wrapper-Connection,] WFModuleThread:
Stopping module class
com.storagetek.diva.proxy.wfm.WFMStateSaverThread...
08/02 16:05:34.411 DEBUG [Wrapper-Connection,] WFModuleThread:
Module class com.storagetek.diva.proxy.wfm.WFMStateSaverThread
stopped.
08/02 16:05:34.411 INFO [Wrapper-Connection,] Stop: All Watch
Folder Monitor modules stopped. WFM state will be saved.
08/02 16:05:34.414 INFO [Wrapper-Connection,] Stop: Watch Folder
Monitor stopped. WFM state was saved.
08/02 16:05:34.414 INFO [Wrapper-Connection,] Launcher: WFM
stopped.
08/02 16:05:34.415 INFO [Thread-9,] Stop: Stopping Watch Folder
Monitor...
08/02 16:05:34.415 DEBUG [Thread-9,] WFModuleThread: Stopping
module class com.storagetek.diva.proxy.wfm.FileMonitorThread...

```

Full Watch Folder Monitor Configuration File (dfm.conf.ini)

This is an example of the full Watch Folder Monitor *dfm.conf.ini* configuration file. This file must be configured and renamed to *dfm.conf* to use Watch Folder Monitor.

```

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

  <!-- DIVA Manager address -->
  <managerConnetion>
    <address host="host" port="8000" 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>foibState.dmp</foibStateFile>

  <!-- File name of the Delete File/Object Information Base dump -->
  <doibStateFile>doibState.dmp</doibStateFile>

  <!-- 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>

<!-- Max 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 range [0..100]. 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
  incomplete. Threshould 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</
  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>SourceDestinationPath</
sourceDestinationDIVAPath>
```

<!-- 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>10000</
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</
deleteParentDirectoryAndContentDirectories>

</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 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
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 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</
deleteParentDirectoryAndContentDirectories>
```

```
</folderConfig>
```

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

  <!-- There are two types of directories: single and file
set. If folder should be single - specify "single" value, 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
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 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.dfm.control.TanukiLaunc
her

# Java Classpath (include wrapper.jar) Add class path elements as
# needed starting from 1
wrapper.java.classpath.1=../../../../../Program/dfm/lib/*.jar

# Java Library Path (location of Wrapper.DLL or libwrapper.so)
wrapper.java.library.path.1=.

# 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

# License Key for Java Service Wrapper
#encoding=UTF-8

```

```
wrapper.license.type=DEV
wrapper.license.id=202004100000007
wrapper.license.licensee=Oracle America, Inc.
wrapper.license.group=Oracle DIVArchive
wrapper.license.dev_application=DIVArchive Drop Folder Manager
wrapper.license.features=pro, 64bit
wrapper.license.upgrade_term.begin_date=2013-01-18
wrapper.license.upgrade_term.end_date=2021-01-18
wrapper.license.key.1=031e-79fe-ce3d-0fc0
wrapper.license.key.2=6dad-ba0d-7dad-e5ff
wrapper.license.key.3=fca0-0e9d-5b50-a54a
wrapper.license.key.4=caef-c841-5222-11b0
-->
</dfmConfiguration>
```