

Oracle® Fusion Middleware

Using Oracle Managed File Transfer



14c (14.1.2.0.0)

F81900-01

December 2024

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Fusion Middleware Using Oracle Managed File Transfer, 14c (14.1.2.0.0)

F81900-01

Copyright © 2020, 2024, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Inside 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, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	xi
Documentation Accessibility	xi
Related Documents	xi
Conventions	xi

1 Understanding Oracle Managed File Transfer

What You Can Do with Oracle Managed File Transfer	1-1
Oracle Managed File Transfer, Oracle SOA Suite, and Oracle B2B	1-3
Oracle Managed File Transfer Functional Use Case Patterns	1-3
Standalone Applications	1-3
SOA Integration	1-4
B2B Integration	1-4
Oracle Service Bus Integration	1-4
Hybrid Integration	1-5
Oracle Managed File Transfer Architecture	1-5
Components of Oracle Managed File Transfer	1-6
Artifacts: Sources, Targets, and Transfers	1-6
Embedded FTP and sFTP Servers	1-6
Monitoring and Reports	1-6
Security	1-7
Utilities	1-7
Repository	1-7
Interfaces	1-8
Oracle Managed File Transfer User Roles	1-8
File Handlers	1-8
Designers	1-8
Monitors	1-8
Administrators	1-9
Installing Oracle Managed File Transfer	1-9
Screen Navigation in Oracle Managed File Transfer	1-9
Design, Monitoring, and Administration Pages	1-9
Using the Left Panel Navigator	1-10

Opening and Closing Dynamic Tabs	1-10
Dragging and Dropping Sources and Targets into Transfers	1-11
Expanding and Collapsing the Dashboard Regions	1-12
Setting Language, Time Zone, and Accessibility Preferences	1-13

2 Designing Artifacts: Transfers, Sources, and Targets

About Designing Transfers	2-1
Getting Ready to Create a Transfer	2-1
Designing End-to-End Flows	2-2
Configuring a Transfer	2-2
Adding a Source and Targets	2-3
Setting Up Content Filters	2-4
Configuring Target-Specific Transfer Settings	2-4
Setting Up Transfer Preprocessing and Postprocessing Actions	2-5
Compression and Decompression Preprocessing Actions	2-6
Find and Replace Preprocessing Action	2-7
Encryption and Decryption Preprocessing Actions	2-8
New Line Conversion Processing Action	2-10
Run Script Preprocessing Action	2-11
ODIInvoke Post-Processing Actions	2-13
Decompression Postprocessing	2-14
Transfer Notification Postprocessing Action	2-14
Duplicating an Existing Transfer	2-17
Creating a Source	2-18
Source Types	2-20
FTP Embedded Source Type	2-20
sFTP Embedded Source Type	2-20
FTP Remote Source Type	2-21
sFTP Remote Source Type	2-23
File Source Type	2-23
SOAP Source Type	2-24
SOA Source Type	2-26
Service Bus Source Type	2-27
B2B Source Type	2-27
ODI Source Type	2-27
Storage Cloud Service Source Type	2-28
OCI Storage Cloud Service Source Type	2-29
WebCenter Source Type	2-30
Setting Up Source Processing Actions	2-31
Compression and Decompression at the Source	2-32
Encryption and Decryption at the Source	2-32

Find and Replace at the Source	2-34
New Line Conversion at the Source	2-35
Run Script Processing at the Source	2-35
Archiving and Deleting Files Before Delivery	2-37
Duplicating an Existing Source	2-37
Creating a Target	2-38
Target Types	2-40
FTP Remote Target Type	2-40
sFTP Remote Target Type	2-42
File Target Type	2-43
SOAP Target Type	2-43
SOA Target Type	2-43
Service Bus Target Type	2-43
B2B Target Type	2-44
ODI Target Type	2-44
Storage Cloud Service Target Type	2-45
OCI Storage Cloud Service Target Type	2-46
WebCenter Target Type	2-47
Moving and Renaming Files After Delivery	2-47
Duplicating an Existing Target	2-48
Setting Up Schedules	2-48
Schedules with Polling Frequency and Minimum Age	2-50
Setting Up Events	2-51
Trigger Events using REST and SOAP Services	2-51
Setting Up Priorities	2-52
Deploying and Testing Transfers	2-53
Deploying a Source, Target, or Transfer	2-53
How to Tell If a Transfer Is Successful	2-53
Locating Received Files	2-54
Importing and Exporting Transfers	2-54

3 Processing Transfers with Custom Callouts

Understanding Custom Callouts	3-1
Creating a Custom Callout: High-Level Steps	3-2
Creating the Code	3-3
Java Code Requirements and Tips	3-3
Java Code for the Newline Conversion Example	3-4
Creating the Callout Definition File	3-5
Locating the Callout Directory	3-6
Running the createCallouts Command	3-6
Testing the Callout	3-7

Adding the Callout to a Source	3-7
Adding the Callout to a Target	3-8
Viewing the Report to Verify the Callout Action	3-8
Updating the Callouts	3-8
Reference Files	3-10
PreCalloutPlugin Interface	3-10
PostCalloutPlugin Interface	3-11
Callout Definition Schema	3-11
PostDeliveryFailureCalloutPlugin Interface	3-13
Validating using Custom Callout	3-16

4 Integrating Oracle Managed File Transfer with Other Products

Compatible Technologies and Integration Strategies	4-1
Managing Domains	4-2
Specifying the Tracking URL	4-3
Integrating with Oracle SOA Suite	4-3
Creating an MFT Reference for a SOA Source	4-4
Creating an MFT Service for a SOA Target	4-4
Interlinked SOA and MFT Reports	4-4
Integrating with Oracle Service Bus	4-5
Creating a Business Service for a Service Bus Source	4-5
Creating a Business Service in Oracle JDeveloper	4-5
Creating a Business Service in the Oracle Service Bus Console	4-5
Creating a Proxy Service for a Service Bus Target	4-6
Creating a Proxy Service in Oracle JDeveloper	4-6
Creating a Proxy Service in the Oracle Service Bus Console	4-7
Integrating with B2B	4-7
Creating a Remote Trading Partner Channel for a B2B Source	4-7
Configuring a B2B Domain for a B2B Target	4-8
Interlinked B2B and MFT Reports	4-8
Integrating with Oracle Data Integrator	4-8
Using the File System and FTP and SFTP for ODI Integration	4-9
Invoking a Web Service for an ODI Source	4-9
Creating a Data Service for an ODI Target	4-10
Integrating with Web Services	4-11
Integrating with Oracle WebCenter Content	4-12
Integrating with Oracle Storage Cloud Service	4-12
MFT WSDL Files	4-13

5 Monitoring Oracle Managed File Transfer

Monitoring Deployed Sources, Targets, and Transfers	5-1
Disabling Sources, Targets, and Transfers	5-2
Reenabling Sources, Targets, and Transfers	5-2
Undeploying Sources, Targets, and Transfers	5-3
Redeploying Sources, Targets, and Transfers	5-3
Versioning Sources, Targets, and Transfers	5-3
Interpreting Artifact Instance Messages	5-4
Monitoring Transfer Flows Using the Main Dashboard	5-4
Interpreting Main Dashboard Metrics	5-5
Using the File Finder	5-6
Monitoring Active Deliveries	5-7
Interpreting Dashboards for All Transfers, Sources, or Targets	5-7
Interpreting Instance Messages	5-7
Interpreting Dashboard Metrics	5-8
Interpreting Single Artifact Transfer, Source, and Target Dashboards	5-9
Interpreting Artifact Instance Messages	5-9
Interpreting Artifact Information	5-10
Interpreting Artifact Dashboard Metrics	5-10
Interpreting Source, Transfer, and Target Reports	5-11
Using the Flow Diagram	5-11
Source Reports	5-12
Transfer Reports	5-13
Target Reports	5-13
Pause and Restart a Transfer	5-14
Diagnose File Delivery Failures	5-14
Resubmit a Transfer	5-14
Bulk Resubmit	5-15
Diagnose Transfer Errors	5-17
Diagnosing Error Messages and Descriptions	5-17

6 Administering Oracle Managed File Transfer

Changing Server Properties	6-1
General Server Configuration Properties	6-2
Performance Properties	6-3
High Availability Properties	6-3
Advanced Delivery Properties	6-3
Runtime MBean Properties	6-4
Importing and Exporting the MFT Configuration	6-7
Increasing Memory to Improve Performance of Large File Transfers	6-8

Oracle WebLogic Server Startup and Shutdown	6-8
Transferring Files Through Firewalls Using the MFT FTP Proxy Server	6-9
Managing Multiple Weblogic Servers and High Availability	6-10
Configuring High Availability	6-10
Preventing Cluster Startup Errors	6-11
Load Balancing in Oracle Managed File Transfer	6-11
Enabling Event Notifications	6-11
Configuring an Email Driver for Notifications	6-12
Configuring an SMS Driver for Notifications	6-13
MFTExceptionQueue	6-14
Configuring Oracle Managed File Transfer Error Processor Queues	6-15
Configuring Oracle Managed File Transfer Loggers	6-17
Oracle Managed File Transfer Component Loggers	6-18
Viewing Oracle Managed File Transfer Log Messages	6-19
Managing Keystores Using the Oracle Managed File Transfer Console	6-20
Using the Keystores Tab	6-20
Using the Keys Tab	6-21
Creating a Key	6-22
Exporting a Key	6-23
Importing a Key	6-23
Updating a Key	6-24
Deleting a Key	6-25
Incrementally Moving Oracle MFT Metadata with Configuration Plans	6-25
Adding a Purge Schedule	6-31
Creating a New Purge Schedule	6-32
Creating a Run Now Purge	6-33
Modifying/Deleting an Existing Purge Schedule	6-33

7 Administering Oracle Managed File Transfer Embedded Servers

About Embedded FTP and sFTP Servers	7-1
Security	7-1
Archiving and Purging Transfers and Files	7-2
Embedded Server Configuration	7-2
Re-configuring the Port	7-2
Path Separators for Remote FTP and sFTP Servers	7-3
Other Embedded Server Settings	7-3
Starting and Stopping Embedded Servers	7-4
Managing Embedded Servers and High Availability	7-5
Supported FTP and sFTP Commands	7-5

8 Oracle Managed File Transfer Security

User Authentication and Authorization	8-1
Configuring Users	8-2
Oracle Managed File Transfer Console Access	8-3
Embedded Server User Access	8-3
Granting Payload Access	8-5
Embedded Server Security	8-6
sFTP (SSH-FTP)	8-6
FTPS (FTP Over SSL)	8-6
Remote SFTP Server Security	8-7
Integrating with Oracle Access Manager 11g for Single Sign-On	8-9
Message Encryption Using PGP	8-9
FIPS 140 Compliance	8-9
Creating an Oracle Managed File Transfer Stripe	8-12
Using Fusion Middleware Control to Create an Oracle Managed File Transfer Stripe	8-12
Using WLST Commands to Create an Oracle Managed File Transfer Stripe	8-13
Managing Keystores Using WLST Commands	8-13
Configuring the SSL Keystore	8-14
Configuring the SSH Keystore	8-15
Configuring the PGP Keystore	8-16
Enable Private Key Passwords for PGP Keys	8-17
Enabling Security Audit Logging	8-20
Using Fusion Middleware Control to Enable Audit Logging	8-20
Using WLST to Enable Audit Logging	8-20
FTP/SFTP Operation in Audit Report	8-21
OWSM Security Policy Attachment	8-21
Using Fusion Middleware Control for Global Policy Attachment	8-22
Managing Policy Credentials	8-22
Creating a Policy Set for a Source	8-23
Creating a Policy Set for a Target	8-24
Using WLST for Global Policy Attachment	8-24
Using the MFT Console for Local Policy Attachment	8-25
Using WLST for Local Policy Attachment	8-26
How the Policy Is Applied at Runtime	8-26
Policies and Artifact Life Cycle Management	8-27
Verifying Policy Registration	8-27
Configuring SSL only Domain for Oracle Managed File Transfer	8-27
Enabling SSL only Domain	8-28

9 Using WLST Commands with Oracle Managed File Transfer

Running WLST Commands	9-1
MFT WLST Command Summary	9-2
Oracle Managed File Transfer EJBs	9-17

Preface

Using Oracle Managed File Transfer describes how to administer Oracle Managed File Transfer, how to design and monitor file transfers, and how to integrate Oracle Managed File Transfer with other applications.

Audience

This document is intended for administrators, designers, deployers, and monitors of file deliveries implemented in Oracle Managed File Transfer. Familiarity with FTP (File Transfer Protocol) and web services is recommended for all users.

For administrators, familiarity with Oracle WebLogic Server and Oracle Database administration is also recommended.

For designers, familiarity with integrating products such as Oracle SOA Suite and Oracle Service Bus is also recommended.

Documentation Accessibility

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

Accessible Access to Oracle Support

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

Related Documents

Refer to the [Oracle Fusion Middleware library on the Oracle Help Center](#) for additional information.

- For Oracle Managed File Transfer information, see Oracle Managed File Transfer.
- For Oracle SOA Suite information, see Oracle SOA Suite.
- For versions of platforms and related software for which Oracle Managed File Transfer is certified and supported, review the [Certification Matrix on OTN](#).

Conventions

The following text conventions are used in this document:

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

1

Understanding Oracle Managed File Transfer

Oracle Managed File Transfer (MFT) is a high performance, standards-based, end-to-end managed file gateway. It features design, deployment, and monitoring of file transfers using a lightweight web-based design-time console. The MFT console includes transfer prioritization, file encryption, scheduling, and embedded FTP and sFTP servers. Security is maintained with security policies such as OWSM. This chapter describes basic Oracle Managed File Transfer concepts.

This chapter includes the following sections:

- [What You Can Do with Oracle Managed File Transfer](#)
- [Oracle Managed File Transfer, Oracle Service Bus, and Oracle B2B](#)
- [Oracle Managed File Transfer Functional Use Case Patterns](#)
- [Oracle Managed File Transfer Architecture](#)
- [Components of Oracle Managed File Transfer](#)
- [Oracle Managed File Transfer User Roles](#)
- [Installing Oracle Managed File Transfer](#)
- [Screen Navigation in Oracle Managed File Transfer](#)



Note:

Screens shown in this guide may differ from your implementation, depending on the skin used. Any differences are cosmetic.

What You Can Do with Oracle Managed File Transfer

You can perform various operations, such as scheduling, file encryption, resubmitting transfers, purging data, and many more such operations by using Oracle Managed File Transfer.

Oracle Managed File Transfer lets you perform the following operations during the transfer process:

- Scheduling
- Resubmitting
- Attaching inline or referencing
- Compression and decompression
- Encryption and decryption
- Archiving, renaming, and deletion
- Purging transfer instances and files
- Pausing and resuming
- Securing with OWSM policies

For more information about resubmitting transfers, see [Resubmit a Transfer](#). For more information about the other operations, see [Designing Artifacts: Transfers, Sources, and Targets](#).

Oracle Managed File Transfer lets you track and troubleshoot file deliveries (transfer instances) based on the following:

- Success, frequency, and failure statistics
- Metrics, recent errors, file finder, and active deliveries
- Error information table
- Active delivery progress table
- Reports for individual deliveries

For more information, see [Monitoring Oracle Managed File Transfer](#).

Oracle Managed File Transfer lets you transfer files to and from many endpoint types:

- File and FTP based endpoints:
 - File: Transfer files from or to directories accessible to the Oracle Managed File Transfer server.
 - FTP Embedded: Transfer files from the embedded MFT FTP (File Transfer Protocol) or FTPS (FTP with Secure Socket Layer) server by copying the file into one of the embedded server directories.
 - sFTP Embedded: Transfer files from the embedded sFTP (Secure Shell FTP or SSH-FTP) server by copying the file into one of the embedded server directories.
 - FTP Remote: Transfer files from or to a remote FTP or FTPS server.
 - sFTP Remote: Transfer files from or to a remote sFTP server.
- SOAP web-services based endpoints:
 - SOAP: Transfer files from or to Simple Object Access Protocol web service endpoints inline or by reference to a folder location.
 - SOA: Transfer files from or to Oracle SOA (Service-Oriented Architecture) web service endpoints.
 - Service Bus (OSB): Transfer files from or to Oracle Service Bus web service endpoints.
 - ODI: Transfer files from or to Oracle Data Integrator web service endpoints.
- B2B based endpoints:
 - B2B: Transfer files from or to Oracle B2B (Business to Business) trading partners.
- Cloud endpoints:
 - Oracle Cloud Service: Transfer files from or to Oracle Cloud Service.
 - Oracle WebCenter Content: Transfer files from or to Oracle WebCenter Content.

For examples of some of these transfers in context, see [Oracle Managed File Transfer Functional Use Case Patterns](#). For full details on how to create these transfers, see [Designing Artifacts: Transfers, Sources, and Targets](#).

Oracle Managed File Transfer, Oracle SOA Suite, and Oracle B2B

Oracle Managed File Transfer, Oracle SOA Suite, and Oracle B2B have overlapping features, but each excels at different file transfer scenarios.

Oracle Managed File Transfer is especially good for:

- Transfer of large files limited in size only by the operating system and capacity of the file system.
- Transfer from a single source with fan-out to many targets.
- Detailed auditing and recording of all transfers.
- Advanced security for transfers.
- Advanced transfer management, such as restart and pause/resume.
- Use of an embedded FTP or sFTP server.

Oracle SOA Suite is especially good for:

- Orchestration or complex integration, such as fan-in from multiple sources.
- Integration with enterprise systems such as messaging or ERP.
- Manual tasks, content based routing, or transformations.

Oracle B2B is especially good for:

- Scenarios in which document format is relevant (for example, HL7, EDI, and so on).
- Scenarios in which additional semantics, such as AS2, are required.

Oracle Managed File Transfer Functional Use Case Patterns

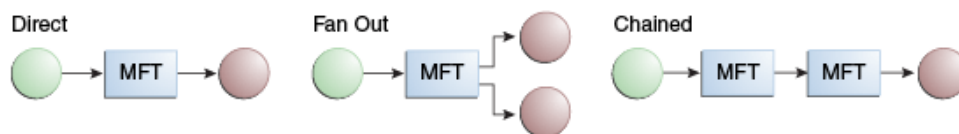
Oracle Managed File Transfer can help integrate applications by transferring files between them in complex use case patterns. There are various common use case patterns and some of them are described in this section.

For full details on how to integrate Oracle Managed File Transfer with other applications, see [Integrating Oracle Managed File Transfer with Other Products](#).

Standalone Applications

Figure 1-1 shows how Oracle Managed File Transfer can transfer files on its own using embedded FTP and sFTP servers and the file systems to which it has access.

Figure 1-1 Standalone Use Case Pattern

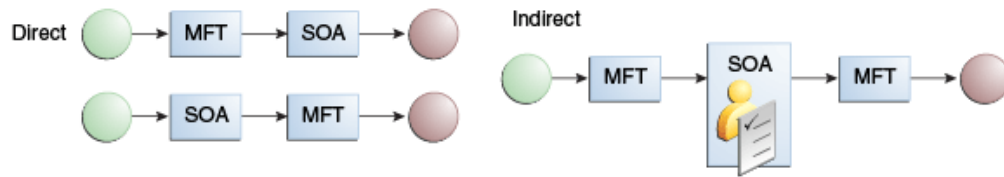


A file transfer can have one or more targets. A multiple target use case pattern is called **fan-out**. The target of one transfer can also use the same endpoint as the source of another transfer, creating a chain.

SOA Integration

Figure 1-2 shows how Oracle Managed File Transfer can integrate with the web service interfaces of SOA applications.

Figure 1-2 SOA Use Case Pattern

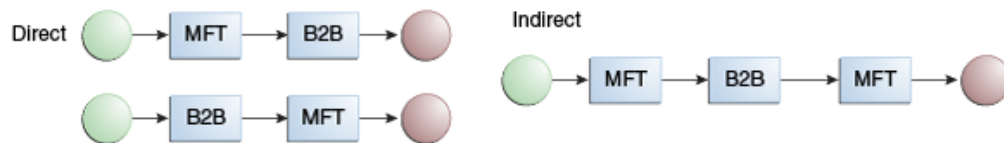


A SOA application can be the source or target of a transfer. A SOA application can also be the common endpoint for the target of one transfer and the source of another.

B2B Integration

Figure 1-3 shows how Oracle Managed File Transfer can integrate with B2B trading partners.

Figure 1-3 B2B Use Case Pattern



A B2B application can be the source or target of a transfer. A B2B application can also be the common endpoint for the target of one transfer and the source of another.

Oracle Service Bus Integration

Figure 1-4 shows how Oracle Managed File Transfer can integrate with Oracle Service Bus web service interfaces.

Figure 1-4 Oracle Service Bus Use Case Pattern

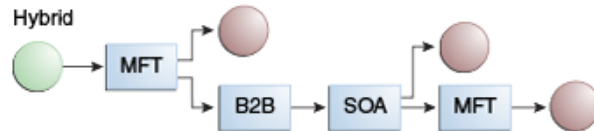


An Oracle Service Bus interface can be the source or target of a transfer. An Oracle Service Bus interface can also be the common endpoint for the target of one transfer and the source of another.

Hybrid Integration

Figure 1-5 shows how Oracle Managed File Transfer can integrate with multiple applications.

Figure 1-5 Hybrid Use Case Pattern

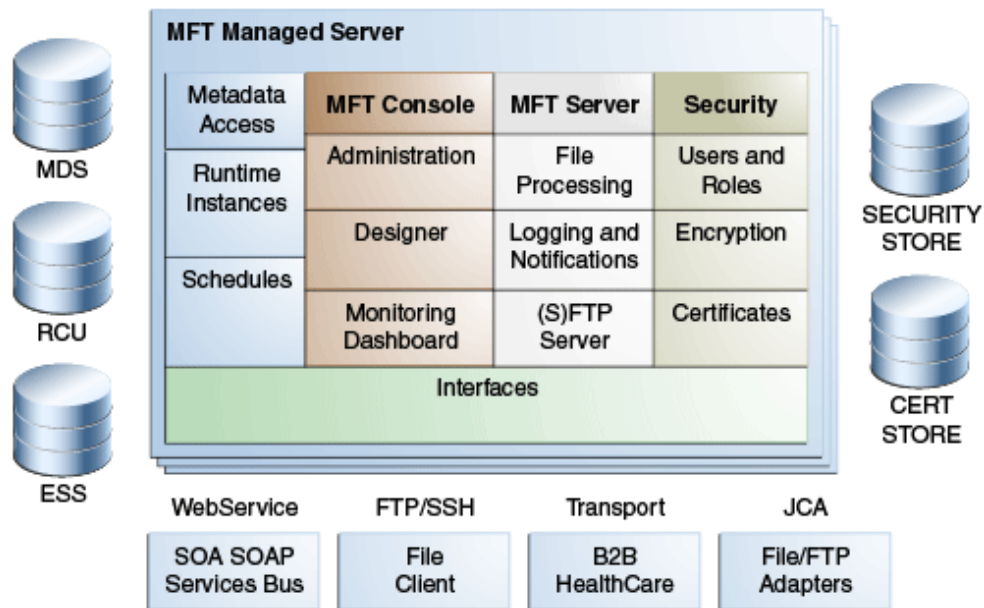


Oracle Managed File Transfer can be one participant in a web of data transfers that includes multiple application types.

Oracle Managed File Transfer Architecture

The main components of Oracle Managed File Transfer includes configuration data, the user-interface console, embedded FTP and sFTP servers, security, and interfaces to various types of file transfer endpoints. Oracle Managed File Transfer can consist of multiple managed servers that provide high availability.

Figure 1-6 Oracle Managed File Transfer Architecture



For details about the console, see [Screen Navigation in Oracle Managed File Transfer](#). For details about the other components, see [Components of Oracle Managed File Transfer](#).

Components of Oracle Managed File Transfer

The components of Oracle Managed File Transfer comprise of artifacts, servers, tools for monitoring, WLST command-line utilities, metadata repository, and various standard interfaces to communicate with source and target endpoints.

Artifacts: Sources, Targets, and Transfers

When you create a file delivery structure using the Designer page of the Oracle Managed File Transfer Console, you create three types of **artifacts**:

- A **source**, which defines an origin of files
- A **target**, which defines a destination of files
- A **transfer**, which associates a source with one or more targets

An **artifact** defines the configuration for parts of a file delivery structure. This is in contrast to a file delivery **instance**, which is an individual file delivery that follows the structure.

Sources and targets can be reused in multiple transfers. When more than one transfer uses the same source, this is called **transfer fan-out**. When a transfer uses more than one target, this is called **target fan-out**. A source and all associated transfers and targets are collectively called a **flow**.

Using various artifact properties, you can define additional file delivery behavior:

- Filters: Files with specific name and extension patterns can be included or excluded.
- Schedules: Transfers can be limited to specific times or time windows.
- Preprocessing actions: Files can be compressed, decompressed, encrypted, or decrypted.
- Postprocessing actions: Files can be decompressed.
- File operations: Files can be archived, renamed, moved, or deleted.

For information about creating artifacts, see [Designing Artifacts: Transfers, Sources, and Targets](#).

Embedded FTP and sFTP Servers

Two servers are embedded in Oracle Managed File Transfer: FTP and sFTP. These two embedded servers can be source artifacts. You can configure various properties of these embedded servers, such as ports, security, and user access to directories. These servers are automatically deployed as part of the WebLogic Server Oracle Managed File Transfer deployment. For more information, see [Administering Oracle Managed File Transfer Embedded Servers](#).

Monitoring and Reports

Oracle Managed File Transfer provides various tools in the Monitoring Dashboard:

- Metrics: real-time displays of transfer status, including failure ratio, payload file size, transfer speed, and the total time of the transfer.
- File Finder: a table that shows either a source or target instance based on the search type.
- Recent Errors: a searchable table of errors that occurred during the transfer.

- Active Deliveries: a table of in-progress, and recently completed file deliveries.
- Reports: detailed information about individual file deliveries from the perspective of the source, transfer, or target.

For more information, see [Monitoring Oracle Managed File Transfer](#).

Security

Oracle Managed File Transfer provides security by:

- Limiting user access to the Oracle Managed File Transfer console
- Limiting user access to embedded FTP and sFTP server directories
- Limiting user access to files in a specific transfer instance
- Securing HTTP endpoint access for web service source types
- Key-based authentication
- FTP over SSL and sFTP transport
- Encryption of files
- Oracle Web Services Manager (OWSM) policies
- FIPS compliance

For more information, see [Oracle Managed File Transfer Security](#).

Utilities

Oracle Managed File Transfer provides WLST command-line utilities for performing many of its functions. Command categories are:

- Artifact management
- Metadata management
- Key management
- Deployment history display
- Transfer management
- Embedded server management
- Callout management
- Event notification management
- Archiving of runtime instances and transferred files
- Purging of runtime instances and transferred files
- Transfer priorities
- Cloud services

For more information, see [Oracle Managed File Transfer Utilities](#).

Repository

Oracle Managed File Transfer stores configuration data in an Oracle Metadata Repository. You can edit, back up, and restore this configuration data. For more information, see [Administering Oracle Managed File Transfer](#).

Interfaces

Oracle Managed File Transfer uses various standard interfaces to communicate with source and target endpoint types, including:

- FTP/SSH: interface for FTP servers, sFTP servers, and file systems
- Web service: interface for SOAP, SOA, Oracle Service Bus, and ODI

Oracle Managed File Transfer User Roles

Oracle Managed File Transfer has four user roles: file handlers, designers, monitors, and administrators. Each of these roles can perform various tasks.

For information about how to create users in these roles, see [User Authentication and Authorization](#).

File Handlers

File handlers perform these tasks:

- Copy files to file transfer staging areas, which are called sources.
- Retrieve files from file transfer destinations, which are called targets.

File handlers have no permissions to access Oracle Managed File Transfer itself. They have the permissions required to access source and target directories and endpoints.

Designers

Designers perform these tasks:

- Create, read, update and delete file transfer sources.
- Create, read, update and delete file transfer targets.
- Create, read, update and delete transfers, which link sources and targets in complete file delivery flows.
- Deploy and test transfers.

Monitors

Monitors perform these tasks:

- Use the Dashboard and reports to ensure that transfer instances are successful.
- Pause and resume lengthy transfers.
- Troubleshoot errors and resubmit transfers.
- View artifact deployment details and history.
- View artifact dependence relationships.
- Enable and disable sources, targets, and transfers.
- Undeploy sources, targets, and transfers.
- Start and stop embedded FTP and sFTP servers.

Administrators

Administrators perform these tasks:

- All file handler tasks
- All designer tasks
- All monitor tasks
- Add other users and determine their roles
- Configure user directory permissions
- Configure the Oracle Managed File Transfer server
- Configure embedded FTP and sFTP servers, including security
- Back up and restore the Oracle Managed File Transfer configuration
- Purge transferred files and instance data
- Archive and restore instance data and payloads
- Import and export metadata

Installing Oracle Managed File Transfer

You can install Oracle Managed File Transfer on its own or in the same Oracle WebLogic Server domain as Oracle SOA Suite.

For more information, see *Preparing to Install and Configure Oracle Managed File Transfer in Installing and Configuring Managed File Transfer*.

Supported system configurations are listed on the Oracle Technology Network web site at <http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>.

Screen Navigation in Oracle Managed File Transfer

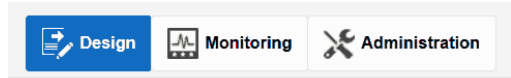
The Oracle Managed File Transfer console allows you to open, close, hide, and expand areas so you can focus on a specific task. Screen Navigation section explains basic console operations.

Design, Monitoring, and Administration Pages

In the top banner of the console, there are links to the top-level pages; which of these pages are accessible depends on your user role. For more information, see [Oracle Managed File Transfer User Roles](#). The top-level pages are:

- **Designer:** Use this page to create, modify, delete, rename, and deploy sources, targets, and transfers.
- **Monitoring:** Use this page to monitor transfer statistics, progress, and errors. You can also use this page to disable, enable, and undeploy transfer deployments and to pause, resume, and resubmit instances.
- **Administration:** Use this page to manage the Oracle Managed File Transfer configuration, including embedded server configuration.

Figure 1-7 Design, Monitoring, and Administration Pages

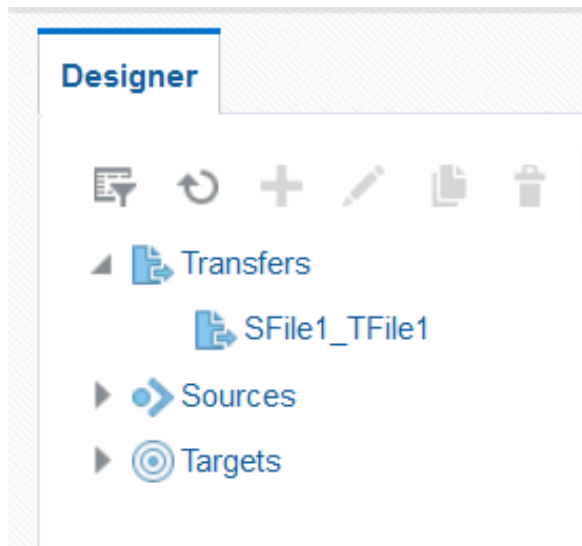


Using the Left Panel Navigator

On each top-level page, a navigation tree in the left panel displays the names of tasks you can perform, most of which open tabs. A right facing arrow to the left of a name indicates hidden subtasks. When you click on the arrow, it faces to the lower right and subtasks are displayed.

For example, the Designer page has three main tasks: Transfer, Source, and Target. An arrow to the left of Transfer indicates that at least one transfer has been created, likewise for Source and Target. Click on the Transfer arrow to display names of transfers.

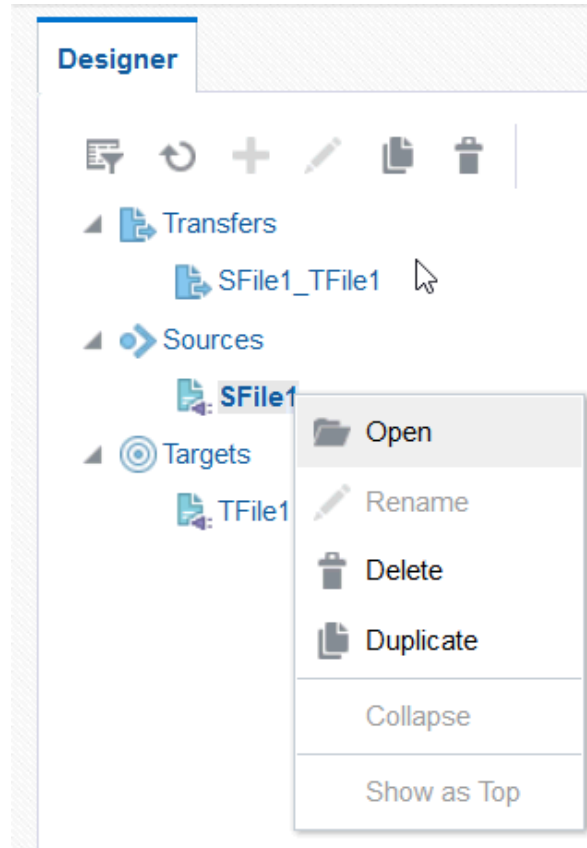
Figure 1-8 Left Panel Navigator on Designer Page



Opening and Closing Dynamic Tabs

To open the tab for a task in the navigation tree, click the task. You can also right click the task and then select the Open button.

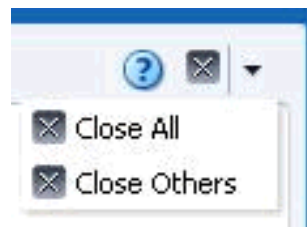
Figure 1-9 Opening a Tab



For example, on the Designer page, to create a new source, click Source, likewise for Transfer and Target. To edit an existing source, transfer, or target, click its name.

To close the active tab, click the Close icon on the right side of the top banner. To close all tabs except the active tab, select Close Others from the drop-down menu to the right of the Close icon. To close all tabs, select Close All from this menu.

Figure 1-10 Options for Closing Tabs



Dragging and Dropping Sources and Targets into Transfers

When you create a transfer and open its tab, **<add source>** and **<add target>** options are displayed. You can drag and drop a source icon and one or more target icons into the respective regions on the target page. You must drag and drop the icon; using the name does not work.

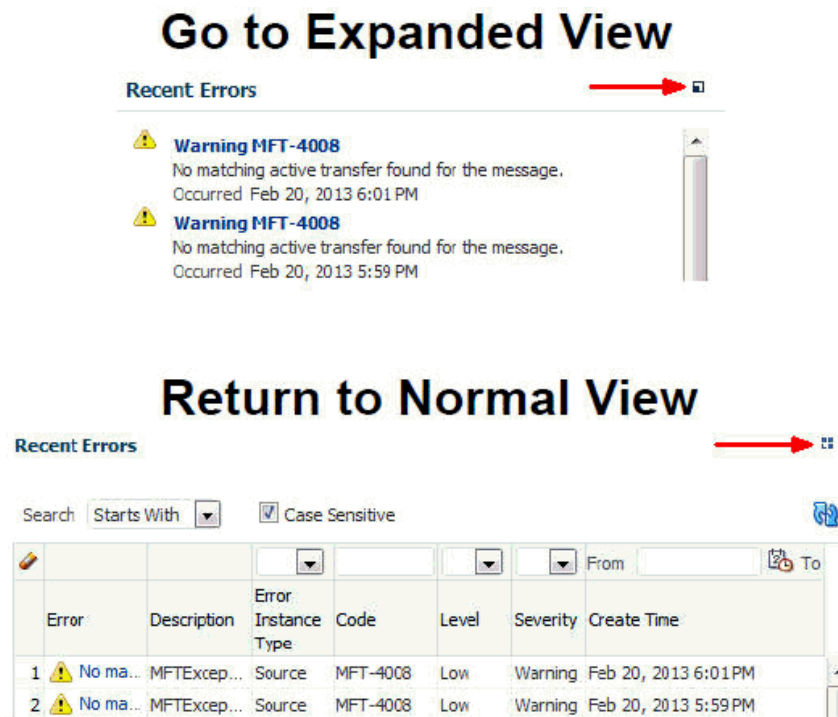
Figure 1-11 Dragging a Source Icon to Add a Source



Expanding and Collapsing the Dashboard Regions

The main tab on the Monitoring page is the Dashboard, which is always open. The Dashboard has four regions: Metrics, File Finder, Recent Errors, and Active Deliveries. In the top right corner of each region is an Expand icon. Clicking this icon expands the region to fill the entire console and displays additional details. To return to the Dashboard, click the Collapse icon in the top right corner of the expanded region.

Figure 1-12 Expanding and Collapsing a Monitoring Region



Setting Language, Time Zone, and Accessibility Preferences

The steps for this process are:

1. On the right side of the top banner of Oracle Managed File Transfer, click **Preferences**.
2. On the left side of the Preferences window, select **Language**.
3. Select a language from the **Language** drop-down list.

No Preference defaults to English.

 **Note:**

You must log out and log in again for a language change to take effect.

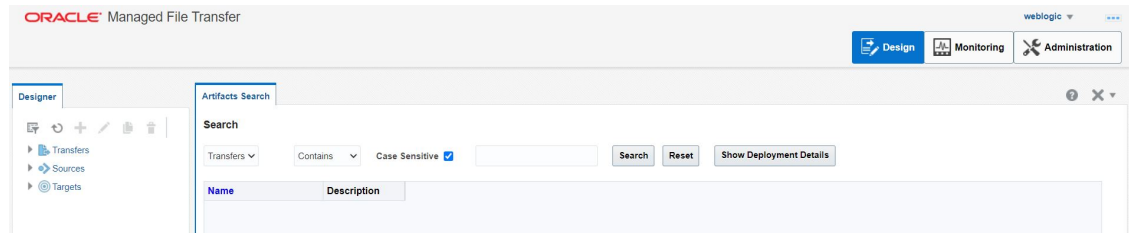
4. Select a time zone for dashboards and alerts from the **Time Zone** drop-down list.
No Preference defaults to coordinated universal time (UTC).
5. On the left side of the Preferences window, select **Accessibility**.
You can also change accessibility settings from the login screen by clicking **Accessibility** in the upper right corner.
6. Select an option from the **Mode Settings** drop-down list:
 - No Preference
 - Enable screen reader mode
 - Disable screen reader modeNo Preference defaults to screen reader mode disabled.
7. Select an option from the **Contrast Settings** drop-down list:
 - No Preference
 - Use high contrast
 - Use normal contrastNo Preference defaults to normal contrast.
8. Select an option from the **Font Settings** drop-down list:
 - No Preference
 - Use large fonts
 - Use normal fontsNo Preference defaults to normal fonts.
9. Click **Apply**, then click **OK**.

Oracle Enterprise Manager Fusion Middleware Control provides accessibility options for the pages on which you monitor and manage Oracle Managed File Transfer. Fusion Middleware Control supports screen readers and provides standard shortcut keys to support keyboard navigation. You can also view the console pages in high contrast or with large fonts for better readability. For information and instructions on configuring accessibility in Fusion Middleware Control, see Using Oracle Fusion Middleware Accessibility Options in *Administering Oracle Fusion Middleware*.

2

Designing Artifacts: Transfers, Sources, and Targets

You can design file delivery structures consisting of sources, targets, and transfers using the Design page in the Oracle Managed File Transfer console.



This chapter includes the following sections:

- [About Designing Transfers](#)
- [Configuring a Transfer](#)
- [Creating a Source](#)
- [Creating a Target](#)
- [Setting Up Schedules](#)
- [Setting up Events](#)
- [Setting Up Priorities](#)
- [Deploying and Testing Transfers](#)
- [Importing and Exporting Transfers](#)

About Designing Transfers

Before designing a file transfer using the Design page in the Oracle Managed File Transfer console, you should design the transfer on paper or a whiteboard.

Getting Ready to Create a Transfer

Transfers are an artifact that links a single source to one or more targets. Transfers can include content filters and other actions that affect the transfer. Before you create a transfer, you should determine the details of what the transfer should do. Consider these specifications:

- The origin location, from which files are transferred, called the source
- The destination locations, to which files are transferred, called the target
- Whether the origin and destinations are file system directories or web service endpoint URLs. The origin and destinations can also be in other applications such as B2B.
- If the file is large, you might want to pass a reference to the web service destination rather than the file.

- Access parameters for the origin and destinations: usernames, passwords, security certificates, and file system permissions
- The file format: binary, XML, or text
- Whether some files must be included or excluded based on format or name
- Whether files must be compressed or decompressed
- Whether files must be encrypted or decrypted
- Whether files must be renamed, moved, archived, or deleted
- Whether files must be scheduled for delivery at specific times or time ranges

 **Note:**

You can use the Artifacts Search tab to see if a source, target, or transfer exists that you can reuse or recreate with modifications. For more information, see Artifacts Search in the *MFT Composer Online Help*.

Designing End-to-End Flows

Sources and targets can be reused in multiple transfers. When more than one transfer uses the same source, this is called **transfer fan-out**. When a transfer uses more than one target, this is called **target fan-out**. A source and all associated transfers and targets are collectively called a **flow**.

You cannot use a source as a target or a target as a source. However, a target and a source can reference the same location. This allows the target of one transfer to be the source of another, creating a transfer **chain**.

In addition to determining the specifications of each transfer, you should map out any fan-outs and chains needed in the overall file delivery structure. For examples, see [Oracle Managed File Transfer Functional Use Case Patterns](#).

Configuring a Transfer

You can create a transfer before or after you create the source and targets. However, you cannot deploy a transfer without a source and at least one target.

The steps to configure a transfer are:

1. Create a transfer in one of these ways:
 - Click **Transfers** in the left pane navigator.
 - Select **Transfers** in the left pane navigator and click the **Create** icon.
 - Right-click **Transfers** in the left pane navigator and select the **Create** command.

The Transfers dialog opens.

2. Type a **Name** for the transfer.
The name can include letters, numbers, dashes, and underscores.
3. Type a **Description** for the transfer.
The description is optional.

4. Click the **OK** button.
A tab for the transfer opens.
To avoid creating a transfer, click **Cancel**.
5. Add a source and one or more targets.
See [Adding a Source and Targets](#).
6. Add content filters.
This is optional. See [Setting Up Content Filters](#).
7. Add users, groups, and roles who can access the transfer payload.
This is optional and applies only to web service pass by reference transfers. See [Granting Payload Access](#).
8. Configure target-specific transfer settings.
This is optional. See [Configuring Target-Specific Transfer Settings](#).
9. Add a schedule.
This is optional. See [Setting Up Schedules](#).
10. Add preprocessing and postprocessing actions such as compression and encryption.
This is optional and applies only to targets. Source actions are added directly in the source artifact. See [Setting Up Transfer Preprocessing and Postprocessing Actions](#).
11. Click the **Save** button.
To undo all changes since the last save, click **Revert**.
12. Click the **Deploy** button after saving.
You can add an optional comment.
If the associated source and target have not been previously deployed, deploying the transfer automatically deploys the associated source and target.
If an existing transfer has most of the desired properties, you can duplicate it. See [Duplicating an Existing Transfer](#).

Adding a Source and Targets

The steps for this process are:

1. Click the arrow to the left of **Transfers** in the left pane navigator.
The transfers are listed. Note that this step might have been completed in the previous section when creating the transfer.
2. Click the transfer name or right-click it and then select the **Open** menu item.
The transfer tab opens.
3. Add a source in one of these ways:
 - Click **add source** if you already created the source. Select the source to add, then select **OK**.
 - Click **create source** to create a new source.
 - Drag and drop a source from the navigation pane.
See [Dragging and Dropping Sources and Targets into Transfers](#) for more information.

4. Add targets in one of these ways:
 - Click **add target** if you already created the target.
Select the target to add in the left column, then click the single arrow icon to move it to the column on the right. You can add more than one target. To select all targets, click the double arrow icon. Click **OK**.
 - Click **create target** to create a new target.
 - Drag and drop a target from the navigation pane.
See [Dragging and Dropping Sources and Targets into Transfers](#) for more information.
5. **Save** and **Deploy** the transfer.

Setting Up Content Filters

Content filters specify file name and extension pattern criteria for transfer. If no content filters are defined, all files at the source endpoint are transferred.

The steps for this process are:

1. Click the arrow to the left of **Transfers** in the left pane navigator.
The transfers are listed.
2. Click the transfer name or select it and then click the **Open** icon.
The transfer tab opens.
3. Click the arrow to the left of Content Filters.
The Content Filters settings are displayed.
4. Select **Wildcard** or **Regular Expression** to determine how the filter string is interpreted.
5. Type a pattern in the text field for the filter.

If you selected **Wildcard**, use * as a wildcard. For example, *.xml specifies that XML files are transferred. To specify text or XML files, you can use *. (xml|XML|txt|TXT). For example, File = "TXT-20200505-XXXX.txt" where XXXX can be any four successive digits.

For more information about regular expressions, see [The Java Tutorials: Regular Expressions](#).

The pattern is for file names only. Filtering on directory names is not supported.

6. Click **add filter** to add another filter.
Another text field is created and given a new number.
7. Repeat these steps for each filter you want to add.
To delete a filter, click the red X to the right of the filter.
8. Use the up and down arrows to the right of each filter to change the filter order.
Lower numbered filters are performed first.
9. **Save** and **Deploy** the transfer.

Configuring Target-Specific Transfer Settings

Depending on the target type, different optional target-specific transfer settings are displayed when you open a transfer tab and click the arrow to the left of a target.

If the target type is File, FTP Remote, or sFTP Remote, the **subfolder** setting is displayed. This adds a transfer-specific subfolder to the target location.

You can override the folder name that is set at the target setting by setting MBean `enableDynamicTargetFoldername` as true, then pass the `TargetFoldername` header via SOAP. A sample SOAP request is shown below:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <ns1:MFTHeader xmlns:ns1="http://xmlns.oracle.com/fmw/mft/soap">
      <ns1:TargetFilename>Order.xml</ns1:TargetFilename>
      <ns1:TargetFoldername>scratch/testD</ns1:TargetFoldername>
      <ns1:ContentIdentifier>Order.xml</ns1:ContentIdentifier>
    </ns1:MFTHeader>
  </soap:Header>
  <soap:Body>
    <ns1:MFTServiceInput xmlns:ns1="http://xmlns.oracle.com/fmw/mft/soap">
      <ns1:InlinePayload>
        <PurchaseOrder>
          </PurchaseOrder>
        </ns1:InlinePayload>
      </ns1:MFTServiceInput>
    </soap:Body></soap:Envelope>
```

Target pre-callout can also be used to add `TargetFoldername` header property:

```
context.getCustomPropertyMap().put("TargetFoldername" , newFolder);
```

Note:

If the Source types that support subfolder option such as File, Embedded FTP/sFTP, Remote FTP/sFTP, OSCS, and WebCenter has **Include Content in Subfolder** option enabled and Target has the **Propagate Source SubFolders** option enabled, the subfolder structure is replicated to the target during the transfer. For example, if there is a file in `/tmp/src/folder/test.txt`, the file gets copied to `/tmp/tgt/folder/test.txt`, the subfolder structure is maintained.

If the target type is an HTTP SOA-based web service type (B2B, SOAP, SOA, Service Bus, or ODI), the following **Delivery Preferences** are displayed:

- **Delivery Method:** Specifies the delivery method: Inline or Reference (default). If Inline, the actual file is sent in the SOA message payload. If Reference, a link to the file is sent.
- **Reference Type:** Specifies the reference type: FTP (default), File, or sFTP. Note that internal and external port numbers can be set in the **Advanced Delivery Properties** area of the Administration **Server Properties** page.
- **Max Inline Size:** Specifies the maximum size in bytes for inline deliveries.

Setting Up Transfer Preprocessing and Postprocessing Actions

After you add a target to a transfer, you can edit the transfer to add preprocessing actions such as compression, decompression (file type only), encryption or decryption, find and replace or new line conversion action. You can also add decompression as a postprocessing action for a target of type File.

You can configure preprocessing for the source; see [Setting Up Source Processing Actions](#).

You can also create custom preprocessing and postprocessing actions; see [Processing Transfers with Custom Callouts](#).

 **Note:**

Postprocessing occurs after file delivery. Therefore, the Active Deliveries and File Finder views in the Dashboard tab on the Monitoring page show different statuses if file delivery succeeds but postprocessing fails. Specifically, the Active Deliveries view displays a Completed status but the File Finder view displays a Failed status.

 **Note:**

If you add the same processing action to a source and a target that uses the source, the action is performed twice. For example, if you add compression to the source and the target, the transferred file is compressed twice.

Decompress action is supported as a pre-processing action for source and target. Multiple file pre-processing decompression is only supported for SOAP, SOA, Service Bus, and ODI target types and delivery preference File /FTP/sFTP. For other target types, a pre-processing decompression error occurs, if a compressed file has multiple entries. Decompression is also supported as post-processing action for File type target which supports multifile decompression.

If you have multiple file decompression, you can workaround by having 2 transfers:

- Transfer 1 : Actual Source -> File target with decompress post-processing
- Transfer 2 : File Source (file in subfolder checked and same folder where transfer 1 file) -> Actual target

 **Note:**

If you copy a binary file to the source location using an FTP client external to Oracle Managed File Transfer, be sure to configure it for binary transfer. Otherwise the file might become corrupted. Processing actions such as compression and encryption might not work properly.

Compression and Decompression Preprocessing Actions

You can compress or decompress a file prior to a transfer delivering it to a target. You can specify either action in the transfer configuration.

Multi-file decompression preprocessing is supported only for SOAP, SOA, Service Bus, and ODI type targets in which the Delivery Method is set to Reference. In this case, the files inside the ZIP file are extracted to a unique random directory, and only a reference to this directory is sent to the target. This directory is listed in the Target Pre-Processing section of the target report. See [Target Reports](#) for more information.

**Note:**

Any processing function added after the multi-file decompression is ignored. If the decompression preprocessing of other types of targets results in multiple files, the decompression action generates an error.

The steps for this process are:

1. Click the arrow to the left of **Transfers** in the left pane navigator.
The transfers are listed.
2. Click the transfer name or right-click it and then select the Open menu item.
The transfer tab opens.
3. Click the arrow to the left of the target.
The target settings are displayed.
4. Click **add pre-processing actions**.
The Pre-Processing Actions dialog opens.
5. Select Compress or Decompress from the All Actions drop-down list.
6. Click **Add to List**.
To remove an action from the list, click the Delete icon to the right of the action.
7. If you selected Compress, select the compression level from the Level drop-down list: Best Compression, Default Compression, or Best Speed. For more information, see the [java.util.zip](#) package, especially the `Deflater` class and the referenced specifications.
8. Click **OK**.
To cancel adding actions, click **Cancel**.
9. **Save** and **Deploy** the transfer.

Find and Replace Preprocessing Action

Use the Find and Replace action to replace a specified text with another text in a file before sending the data to the target. You can perform multiple find and replace actions on a file.

If you are using the Find and Replace action on a specific target, then you may set the processing action at the specific target level. If you are setting the action on multiple targets with the same source, then add the action at the source level. See section [Find and Replace at the Source](#).

To Find and replace action:

1. Click the arrow to the left of **Transfers** in the left pane navigator.
The transfers are listed.
2. Click the transfer name or right-click it and then select the Open menu item.
The transfer tab opens displaying the General Information and Transfer Definitions.
3. Click the Target Definitions tab.
4. Click **Pre-Processing Actions** for Targets.
The Pre-Processing Actions dialog opens.

5. Select **Find and Replace** from the All Actions drop-down list.
6. Click **Add to List**. The Find and Replace Processing action is added in the Selected Actions list.

To remove an action from the list, click the Delete icon to the right of the action.
7. In the **Find** field, add the text that you want to find and in the **Replace with** field, enter replacement text. The find and replace action is case-sensitive. You can add multiple Find and Replace text in a single Find and Replace Action by clicking the Add + icon.
8. To add another replacement text, click the Add + icon.
9. You can add multiple Find and Replace preprocessing actions for each target. To add another Add and Replace action, click Find and Replace from the All Action list.

Find text is a mandatory field. Hence cannot be left empty. If the fields are empty, you get an error.
10. Click OK. Or click Cancel to cancel the action.

After the successful transfer, the report can be seen in the Monitor Dashboard. For more information see [Monitoring Deployed Sources, Targets, and Transfers](#).

Encryption and Decryption Preprocessing Actions

You can encrypt or decrypt a file prior to transfer. You can specify only one encryption or decryption algorithm in the transfer configuration. Along with PGP algorithm, MFT supports PGP signatures. You can generate a signed and encrypted payload and validate the signature when decrypting, this can be done at the artifact level.



Note:

PGP keystores must be configured and certificates must be imported before you add an encryption or decryption action.

If a payload is encrypted by a PGP tool outside of MFT using a key length or algorithm that is restricted, MFT decryption fails. These restrictions are mostly specified at the JRE level in the `JAVA_HOME\jre7\lib\security` directory.

The steps for this process are:

1. Click the arrow to the left of **Transfers** in the left pane navigator.

The transfers are listed.
2. Click the transfer name or right-click it and then select the Open menu item.

The General Information and Target Definitions tab opens.
3. Click the arrow to the left of the Target Definition tab.

The target settings are displayed.
4. On the Target details, click **Pre-processing actions**.

The Pre-Processing Actions dialog opens.
5. Select PGP Encryption or PGP Decryption from the All Actions drop-down list.
6. Click **Add to List**. The selected action appears in the Selected Actions list.

To remove an action from the list, click the Delete icon to the right of the action.

7. If you selected **PGP Encryption**, select values from the Encryption Alias, Armored, Encryption Algorithm, and Signing key alias drop-down lists:
 - Encryption Alias: the public key alias for encryption. For more information about key aliases, see [Configuring the PGP Keystore](#).
 - Armored: Binary or ASCII. Use ASCII if non-printing characters might be stripped in transit.
 - Encryption Algorithm: Select from the following supported algorithms:

 **Note:**

If no algorithm is selected, global algorithm settings apply.

- Default
 - Triple-DES
 - CAST5 – set as default algorithm
 - Blowfish
 - DES
 - AES-128
 - AES-192
 - AES-256 – set as default algorithm if FIPS mode is enabled
 - Twofish
- Signing key alias: Select from the list of imported private signing keys.
8. If you selected **PGP Decryption**, select the Decryption Alias from the drop-down list. This is the private key alias for decryption. For decrypting, the signature must be imported in the PGP keystore.

For more information about key aliases, see [Configuring the PGP Keystore](#).

9. Click **OK**.
To cancel adding actions, click **Cancel**.
10. **Save** and **Deploy** the transfer.

After successful transfer, you can monitor the result in the Monitor dashboard. See [Monitoring Deployed Sources, Targets, and Transfers](#) and [Transfer Reports](#).

Changing Encryption Algorithm for PGP

To change encryption algorithm for PGP, follow these steps:

1. **Step A**
 - a. Log in to Oracle Enterprise Manager Fusion Middleware Control.
 - b. Expand the SOA node and select the soa-infra node.
 - c. From the **SOA Infrastructure** menu, choose **Administration** > **System MBean Browser** . The System MBean Browser page is displayed.
 - d. Under **Application Defined MBeans**, expand the server - oracle.as.soainfra.config node. For example, oracle.as.mftinfra.config node.

- e. Expand the Server: soa_server1 node For example, mft_server1 node.
 - f. Expand the **MFTConfig** node.
 - g. Click the MFT **MBean**. The properties of the MBean are displayed in the right pane.
 - h. Click the **Operations** tab.
 - i. Click **addProperty** operation in the list. Enter values for the key, value, and optional comments.
 - i. Set Key value to "pgpEncryptionAlgorithm"
 - ii. Set value. For example, set value as "2" for Triple DES, and click **Invoke**.
2. Step B
- a. In MFT, to change the encryption algorithm for PGP, set the "pgpEncryptionAlgorithm" MBean for MFT. This Mbean accepts int value and different values for the supported algorithms are listed below:
 - i. Triple DES = 2;
 - ii. CAST5 = 3;
 - iii. Blowfish = 4;
 - iv. DES = 6;
 - v. AES 128 = 7;
 - vi. AES 192 = 8;
 - vii. AES 256 = 9;
 - viii. Twofish = 10;
 - b. In FIPS mode, when MBean property PGPEncryptionAlgorithm is not defined, then the default algorithm is AES 256. Supported algorithms are:
 - i. AES 128
 - ii. AES 192
 - iii. AES 256
 - c. In non-FIPS mode, when MBean property PGPEncryptionAlgorithm is not defined, then the default algorithm is CAST5. Supported algorithms are:
 - i. Triple DES
 - ii. Blowfish
 - iii. DES
 - iv. AES 192
 - v. AES 128
 - vi. AES 256
 - vii. Twofish

New Line Conversion Processing Action

Use the New Line Conversion preprocessing action to convert new line characters for different operating systems. The New line conversion action converts the new line character to the

specified operating system specific new line character. New line conversion action can be added at source processing action or target preprocessing action.

The steps for this process are:

1. Click the arrow to the left of Transfers in the left pane navigator.
The transfers are listed.
2. Click the transfer name or right-click it and then select the Open menu item.
The General Information and Target Definitions tab opens.
3. Click the arrow to the left of the Target Definition tab.
The target settings are displayed.
4. On the Target details, click **Pre-processing actions**.
The Processing Actions dialog opens.
5. Select **New line conversion** from the drop-down list.
New line conversion is displayed in Selected Actions.
6. In the **Type** field, select from the list:
 - DOS to Unix
 - Unix to DOSTo delete the New line conversion, click the Delete icon to the left of the action.
7. Click OK.
To cancel, click **Cancel**.

Run Script Preprocessing Action

Use the Run Script preprocessing action for sources and targets to execute any script or command like shell commands, perl commands or bat files on a file before delivering it to the target. You can execute any custom commands such as Virus Scan, external encryption file processing, add new endpoints, enable REST, notify or validate inside the script. You may run the script to modify the payload by replacing certain words or add or validate signature information during encryption and decryption.

Example of a script adding a header:

```
#bin/sh
echo "file generated by script copy"
while read line; do
echo ${line}
done
]
```

Example of a script for compression:

```
#bin/sh
gzip -c
```

Example of a error script :

```
#bin/sh
regexA="*.bak"
regexB="*.BAK"
if [[ "$fileName" == $regexA || "$fileName:" == $regexB ]]; then
    echo "Processing backup file"
else
    echo "Input file[$fileName] is not valid backup file!" 1>&2
    exit 1
fi
```

In the error script example, considering transfer is created to move backup (*.bak) files from source to target, if transfer file is not a backup (*.bak) file, transfer will fail with error "Input file[f2] is not valid backup file."

There are pre-defined variables which can be used in the script, for example `fileName` is a pre-defined variable.

To add Run Script pre-processing action:

1. Click the arrow to the left of Transfers in the left pane navigator.
The transfers are listed.
2. Click the transfer name or right-click it and then select the Open menu item.
The General Information and Target Definitions tab opens.
3. In Target Definitions tab, on the Target details, click **Pre-Processing Actions**.
The Pre-Processing Actions dialog opens.
4. Select **Run Script** action from the **All Actions** drop-down list and click **Add to list**.
It appears in the Selected Actions list.
5. Enter the following details:
 - **Command:** enter the path of the script you want to execute. The script needs to be an executable file. For example, `/home/user/echo.sh`
 - **Timeout:** Specify the timeout value, if the execution of the script takes more than specified time it will stop the execution of the script. If there is an error in the transfer, the error is reflected in the MFT monitoring dashboard, after diagnosis it can be resubmitted.
 - **Read Input Payload:** select the checkbox if you want the script (provided in Command field) to read the input payload before transfer. By default, the checkbox is selected.
 - **Use Script Generated Payload:** select the checkbox if you want to modify the existing payload with the output generated by the script. When unchecked, the script is executed without modifying the payload.
 - **New File Extension:** Specifies the extension that is added to the new file after processing is done. For example, when using compression action, you may want to modify the name to `<filename>.zip`, then you will set 'zip' as the new file extension.
6. Click **Add or Update Script Variable** if you want to update, add or modify any parameter or variable in the script. There are pre-configured runtime parameters (*filename, payload directory, filesize, targetname, sourcename, useGeneratedFileFromScript*) which are updated and passed to script.
 - a. To add a variable to the script, click the Add row icon +.

- b. Enter the **Name** of the variable and the **Value**.
 - c. If the variable value must be encrypted, check the box against **Is Credential**
 - d. To delete a variable, click the Delete icon next to the Value field.
 - e. Click OK.
7. Click **OK** to save the action or **Cancel** to cancel the action.

ODIInvoke Post-Processing Actions

This post processing function is used to configure the ODIInvoke web service and is invoked after delivering the payload to the JCA bindings configured at the ODI target. The ODIInvoke service triggers an ODI flow to retrieve the payload from the JCA binding target type configured at the ODI target. The MFT message is marked as complete once payload delivery is completed and the `odiInvoke` service is invoked.

The MFT ODI target configures the JCA binding types along with the existing SOAP method of message delivery. You can configure the following binding types in the ODI target:

- File- Transfer file via File
- FTP Remote- Transfer file via FTP
- sFTP Remote- Transfer file via sFTP
- SOAP- Transfer file via ODI SOAP DataService

After selecting one of the JCA binding types, you must configure the required parameters for the JCA targets. Once a binding type is selected for an ODI target, you can't change the binding type, but you can continue modifying the parameters of the current JCA target binding.

When you add an ODI target with JCA binding to a transfer, a post processing function, `OdiInvokeWebService`, must be configured at the target. You need to configure the `OdiInvoke` service URL, action, and other required parameters in this post processing function.

In the OdiInvoke post processing function, fields are exposed to configure the `odiInvoke` service URL, port, operation, and other parameters exposed below:

- Request ScenarioRequestType
- ScenarioName string
- ScenarioVersion string
- Context string
- Synchronous boolean
- SessionName string
- Keywords string
- Variables VariableTypeArray Size
- Variables VariableType Name and Value string
- Debug DebugType

For Debug to work correctly, explicitly select the Debug checkbox. If you check any other option (under Debug such as BreakOnError) without selecting the Debug checkbox, it may not work correctly.

During MFT message processing for an ODI target of a binding type JCA, the payload is moved to the JCA location configured at the target. The message is constructed in the format expected by the `odiInvoke` service by reading all the configuration parameters. Then the URL

configured at the ODI target is invoked and the request message is constructed. Once MFT is able to invoke the ODI target successfully, MFT's message is marked as *COMPLETED*.

Decompression Postprocessing

You can decompress a file after transfer only if the target type is File. You can specify this action in the transfer configuration.

Multi-file decompression postprocessing is supported. In this case, the decompressed files are extracted to a directory under the target location having the name of the ZIP file without the extension. For example, if the target location is `/tmp/mft` and the transferred file with multiple entries is `order.zip`, decompressed files are extracted to `/tmp/mft/order`.

Note:

Any processing function added after the multi-file decompression is ignored.

The steps for this process are:

1. Click the arrow to the left of **Transfers** in the left pane navigator.
The transfers are listed.
2. Click the transfer name or right-click it and then select the Open menu item.
The transfer tab opens.
3. Click the arrow to the left of the target.
The target settings are displayed.
4. Click **add post-processing actions**.
The Post-Processing Actions dialog opens.
5. Select Decompress from the All Actions drop-down list.
6. Click **Add to List**.
To remove an action from the list, click the Delete icon to the right of the action.
7. Click **OK**.
To cancel adding actions, click **Cancel**.
8. **Save** and **Deploy** the transfer.

Transfer Notification Postprocessing Action

The Transfer Notification postprocessing action is used to notify users on successful payload transfer. You can specify which channel—email or SMS—to use for sending the notification. You can configure the format of the email/message, file name pattern, and minimum file size to notify.

Note:

MFT supports plain text in successful transfer notification template.

Ensure that the driver properties `usermessagingdriver-email (mft_server1)` in User Messaging Service (UMS) are set. For more information, refer to Oracle User Messaging Service Drivers.

To add Transfer Notification as a postprocessing action:

1. Click the arrow to the left of **Transfers** in the left pane navigator.
The transfers are listed.
2. Right-click the transfer name and select **Open**.
The transfer tab opens.
3. Click the arrow to the left of the Target Definition.
The target settings for Source and Targets are displayed.
4. Click **add post-processing actions**.
The Post-Processing Actions dialog opens.
5. Select **Transfer Notification** from the All Actions drop-down list.
6. Click **Add to List**.
To remove an action from the list, click the Delete icon to the right of the action.
7. Provide the following details on the Transfer Notification fields:
 - **Template File:** (Optional) Specifies an e-mail template file location to be sent as part of transfer notification. For example: `/u01/data/mft/notify.eml`.

When a variable is specified in the Template File field (for example `%%FILENAME%%`) Oracle MFT takes the File name from the run-time session. The pre-seeded variables you can use to create a template are `FILENAME`, `DATE`, `TRANSFERURL`, `SOURCENAME`, `USER`, `FILESIZE`, `TARGETFILENAME`, `TARGETNAME`, `TRANSFERNAME`, `TARGETENDPOINTREF`, `TARGETFILESIZE`.

Note that `To`, `CC`, and `BCC` fields are optional parameters, however `Subject` and `Body` are mandatory parameters. If the **Template File** field is blank, the following default template file is used:

```
CC= abc.xyz@oracle.com; user@oracle.com; test.user@oracle.com
```

```
BCC= test@oracle.com; new.user@oracle.com
```

```
Subject= File %%FILENAME%% Successfully Processed
```

```
Body= The file %%FILENAME%% uploaded by %%USER%% of size %%FILESIZE%% from the source %%SOURCENAME%% was successfully processed on %%DATE%% by Oracle MFT. You can optionally view the %%TRANSFERURL%% details of the transfer from the console.
```

- **Minimum File Size:** Specifies the minimum file size in MB for sending notification.
- **Pattern Type:** Specifies how the filter string is interpreted: Wildcard (the default) or Regular Expression.

If you selected Wildcard, use `*` as a wildcard. For example, `*.xml` specifies that XML files are transferred. To specify text or XML files, you can use `*(.xml|XML|txt|TXT)`.

For example, `File = "TXT-20100505-XXXX.txt"` where `XXXX` can be any four successive digits.

```
Regular Expression = "XT-20100505-\\d{4}\\..txt"
```

- **File Name Pattern:** (Optional) Specifies the file name pattern for sending notification.

- **Add or Update Contacts to Notify:** Select this option if you want to add user contacts to notify. When clicked, Search Contacts dialog opens.
- **Search Contacts:** You can search and add contacts by type - User, External, or Group.

Before you add users in the **Add or Update Contact Users to Notify**, you have to create and configure new contact users using the WLST command. You can create new users or you can use existing WebLogic users and User groups.

 **Note:**

You can configure only internal contacts using the WLS console, which uses the LDAP user settings. You can use the WLST commands to configure both internal and external contacts.

To create new external users, use the WLST command, `createContact`. To use existing Weblogic users or user groups, use the WLST command, `createUserContact` or `createUserGroupContact`. These users are internal and the email address and phone number come from the LDAP user setting. For more information, refer to [Enable Event Notification](#).

Before you can run WLST commands, you must start WLST and connect to the Oracle WebLogic Server managed server dedicated to Oracle MFT. For more information, refer to [Running WLST Commands](#).

To create user contacts:

- To create **User** contact (internal user), use the WLST command:

```
createUserContact('weblogic','Email')
```

Attributes	Description	Syntax	Example
wls:/soainfra/serverConfig/>help('createUserContact')	Create a new user contact, which can be used for event notification. Shortcut for this command is 'crtUCont'.	<code>createUserGroupContact (userGroupName, deliveryChannel) userGroupName: user group name deliveryChannel (optional): possible values are Email/SMS. If not specified, it will use the user preferred delivery channel configured in the WebLogic user store.</code>	wls:/mydomain/serverConfig>createUserContact('user1')wls:/soainfra/serverConfig/>

To create a **Group** user (internal user), use the WLST command:

```
createUserGroupContact('usergroup')
```

Attributes	Description	Syntax	Example
wls:/soainfra/serverConfig/>help('createUserGroupContact')	Create a new user group contact, which can be used for event notification. The shortcut for this command is 'crtUGCont'.	createUserGroupContact (userGroupName, deliveryChannel) userGroupName: user group name deliveryChannel (optional): possible values are Email/SMS. If not specified, it will use the user preferred delivery channel configured in the WebLogic user store.	wls:/mydomain/serverConfig>createUserGroupContact('userGroup1')

To create a new **External** contact, use the WLST command:
createContact('Email','abcd@efgh.com')

Attributes	Description	Syntax	Example
wls:/soainfra/serverConfig/>help('createContact')	Create a new contact, which can be used for event notification. The shortcut for this command is 'crtCont'.	createContact(ContactType,value) ContactType: Email SMS value: Value for the contact type. For example, email ID or phone number.	wls:/mydomain/serverConfig>createContact('Email','abcd@efgh.com')

- b. Go to the WebLogic Server Administration Console and add the email/contact number for the user.

In the WebLogic Server Administration Console, navigate to **Home >Summary of Security Realms > myrealm > Users and Groups > user** and update the channel of communication. Enter the email address and the phone number for SMS.

- c. Search for users in Transfer Notification dialog and associate with the notification action.

The search result is corresponding to the user type. For example, if you search for External users, only External users are listed, User and Group users are not listed.

8. Click **Add** after adding or modifying the contacts. The added contacts are listed in the **Selected Contacts for Notification** text box.
9. Click **OK** to continue.

Duplicating an Existing Transfer

You can create a new file transfer by copying an existing one. The new transfer references the same source and targets as the copied transfer.

The steps for this process are:

1. Duplicate a transfer in one of these ways:
 - Select the transfer to copy and then the **Duplicate** icon in the left pane navigator.
 - Right-click the transfer to copy in the left pane navigator and select the **Duplicate** command from the pop-up menu.

The Duplicate Transfer dialog appears.

2. Type a **Name** for the transfer.

The name can include letters, numbers, dashes, and underscores.

3. Click the **Create** button.

A tab for the transfer opens, providing additional settings you can edit. For more information about these settings, see these sections:

- [Adding a Source and Targets](#)
- [Setting Up Content Filters](#)
- [Granting Payload Access](#)
- [Configuring Target-Specific Transfer Settings](#)
- [Setting Up Schedules](#)
- [Setting Up Transfer Preprocessing and Postprocessing Actions](#)

To avoid creating a transfer, click **Cancel**.

4. Click the **Save** button after editing.

To undo all changes since the last save, click **Revert**.

5. Click the **Deploy** button after saving.

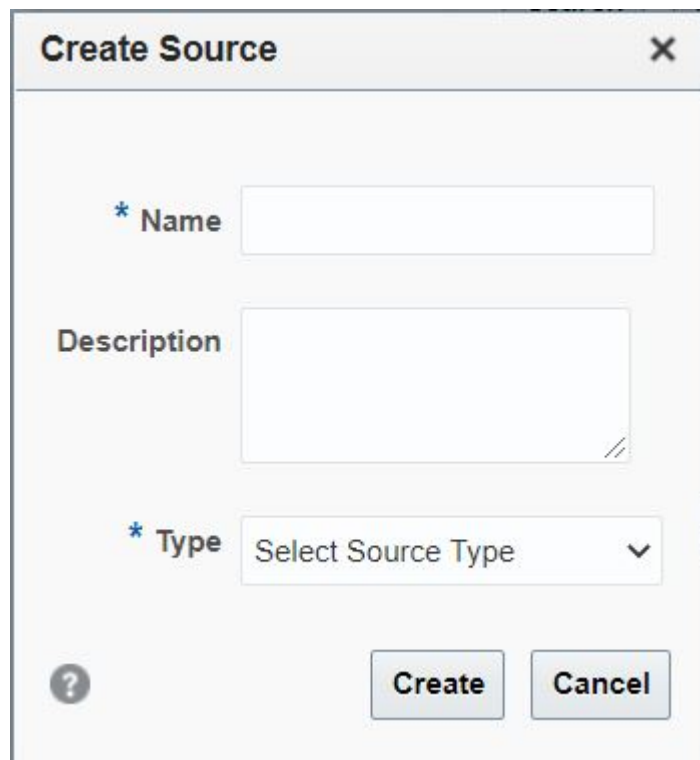
Deploying a transfer deploys the associated source and target automatically.

Creating a Source

You can create a source with a minimum number of settings. After you create a source, you can edit to add more settings to it.

To create a source:

1. In the Designer left pane navigator, click **Sources** to open the Create Source dialog.



2. Enter a **Name** for the source.

The name can include letters, numbers, dashes, and underscores.

 **Note:**

For a SOA or SOAP source, file names must not have spaces.

3. Optionally, enter a **Description** for the source.

4. Select the source **Type**.

This selection determines the other settings that appear. See [Source Types](#).

5. Type a value for the source location. For most source types, this is either:

- The **Folder** setting, which specifies a file system directory. Ensure that the folder name does not exceed 60 characters.
- The **URL** setting, which specifies a web service endpoint.

The B2B source type has no source location setting in the Sources dialog. You must provide the source location after creating the source.

6. Enter values for the remaining required settings, indicated by blue asterisks.

7. Click **Create**.

A tab for the source opens, providing additional settings you can edit. For more information about these settings, see the source type under [Source Types](#), [Setting Up Schedules](#), [Setting Up Source Processing Actions](#), and [Archiving and Deleting Files Before Delivery](#).

8. After editing, click **Save**.

To undo all changes since the last save, click **Revert**.

9. (Optional) Click **Deploy**.

 **Note:**

Deploying a transfer deploys the associated source and target automatically.

If an existing source has most of the desired properties, you can duplicate it. See [Duplicating an Existing Source](#).

Source Types

Oracle Managed File Transfer provides the following source types:

- [FTP Embedded Source Type](#)
- [sFTP Embedded Source Type](#)
- [FTP Remote Source Type](#)
- [sFTP Remote Source Type](#)
- [File Source Type](#)
- [SOAP Source Type](#)
- [SOA Source Type](#)
- [Service Bus Source Type](#)
- [B2B Source Type](#)
- [ODI Source Type](#)
- [Storage Cloud Service Source Type](#)
- [OCI Storage Cloud Service Source Type](#)
- [WebCenter Source Type](#)

FTP Embedded Source Type

Using the FTP Embedded source type means uploading files to the FTP server embedded by Oracle Managed File Transfer, which transfers the files. The only required setting is **Folder**, which specifies the embedded FTP server directory from which to transfer files.

 **Note:**

Files present in the embedded FTP source directory before the source is deployed or enabled are ignored. Only files uploaded to the directory after deployment or enabling are picked and transferred.

For information about additional settings available after you create the source, see [Source—FTP Embedded](#) in the *MFT Composer Online Help*.

sFTP Embedded Source Type

Using the sFTP Embedded source type means uploading files to the sFTP server embedded by Oracle Managed File Transfer, which transfers the files. The only required setting is **Folder**, which specifies the embedded sFTP server directory from which to transfer files.

 **Note:**

Files present in the embedded sFTP source directory before the source is deployed or enabled are ignored. Only files uploaded to the directory after deployment or enabling are picked and transferred.

For information about additional settings available after you create the source, see Source—sFTP Embedded in the *MFT Composer Online Help*.

FTP Remote Source Type

Using the FTP Remote source type means transferring files from an FTP server outside of Oracle Managed File Transfer.

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see Source—FTP Remote in the *MFT Composer Online Help*.

Element	Description
Host Name	Specifies the host name.
Folder	If creating a source, specifies the location of files to be transferred as a directory in a file system. If creating a target, specifies the location to which files are transferred as a directory in a file system.
User	Specifies the user who has access to the source or target. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for user names that start with \$. This is only for a leading \$. If there are other \$s in the user name, do not add more backslashes. Example: for \$john\$smith, enter the password as \ \$john\$smith.
Password	Specifies the user password. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for passwords that start with \$. This is only for a leading \$. If there are other \$s in the password, do not add more backslashes. Example: for \$xyz\$123, enter the password as \ \$xyz\$123.
Confirm Password	Confirms the user password.
Control Port	Specifies the port for the source or target.
SSL	Specifies the use of SSL if checked. This is optional.
Implicit SSL	Specifies the use of implicit SSL if checked. This is optional.

 **Note:**

If you choose List Parser Key as Windows, the recent date and default date format is automatically changed to MM-dd-yyyy HH:mm which is the only supported format in Windows. However, for connecting to Windows (s)FTP Servers, you need to change the List Parser Key to Windows, and have the recent date and default date format configured to MM-dd-yyyy HH:mm format.

FTP Remote Source Advance Properties for MVS Transfers

When creating a FTP Remote source type for MVS Mainframe systems, you need to select MVS as List Parser Key in advance properties.

For MVS FTP response formats, MVS can be configured to use HFS (Unix style) response or MVS native response formats.

FTP Remote Source Settings for MVS Transfers

Configure the FTP Remote source type as shown in figure below:

Note:

The following properties must be selected in the format listed below:

- Change Directory="true"
- Content Folder is a mandatory and must be in the format: "FOLDER."
- FTP Path Separator = ""
- Absolute Path Begin = ""
- List Parser Key = MVS

sFTP Remote Source Type

Using the sFTP Remote source type means transferring files from an sFTP server outside of Oracle Managed File Transfer.

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see *Source—sFTP Remote* in the *MFT Composer Online Help*.

Element	Description
Host Name	Specifies the host name.
Folder	If creating a source, specifies the location of files to be transferred as a directory in a file system. If creating a target, specifies the location to which files are transferred as a directory in a file system.
User	Specifies the user who has access to the source or target. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for user names that start with \$. This is only for a leading \$. If there are other \$s in the user name, do not add more backslashes. Example: for \$john\$smith, enter the password as \john\$smith.
Password	Specifies the user password. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for passwords that start with \$. This is only for a leading \$. If there are other \$s in the password, do not add more backslashes. Example: for \$xyz\$123, enter the password as \xyz\$123.
Confirm Password	Confirms the user password.
Control Port	Specifies the port for the source or target.
Authentication Type	Specifies the type of user authentication: Password, Public Key, or Multi Level.
Multi Level Authentication Order	Specifies the order of execution of authentication for multilevel authentication type. This property defines whether password or public key authentication is validated first.

Note:

If you choose List Parser Key as Windows, the recent date and default date format is automatically changed to `MM-dd-yyyy HH:mm` which is the only supported format in Windows. However, for connecting to Windows (s)FTP Servers, you need to change the List Parser Key to Windows, and have the recent date and default date format configured to `MM-dd-yyyy HH:mm` format.

File Source Type

Using the File source type means transferring files from the local file system or from a network-attached system. The only required setting is **Folder**, which specifies the directory from which to transfer files. This directory must be accessible from Oracle Managed File Transfer.

For information about additional settings available after you create the source, see *Source—File* in the *MFT Composer Online Help*.

Oracle Managed File Transfer uses the same file adapter used by Oracle SOA Suite.

SOAP Source Type

Using a SOAP source type means transferring files from a web service endpoint. The only required setting is **URL**, which specifies the web service endpoint from which to transfer files.

For information about additional settings available after you create the source, see *Source—SOAP* in the *MFT Composer Online Help*.

To capture the filename or store the target filename in source, you need to pass the filename for inline payloads, as shown in the SOAP request below:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <ns1:MFTHeader xmlns:ns1="http://xmlns.oracle.com/fmw/mft/soap">
      <ns1:TargetFilename>Order.xml</ns1:TargetFilename>
      <ns1:ContentIdentifier>Order.xml</
ns1:ContentIdentifier>
      <ns1:file.name>OrderIn.xml</ns1:file.name>
    </ns1:MFTHeader>
  </soap:Header>
  <soap:Body>
    <ns1:MFTServiceInput xmlns:ns1="http://xmlns.oracle.com/fmw/mft/
soap">
      <ns1:InlinePayload>
        <PurchaseOrder>
          </PurchaseOrder>
        </ns1:InlinePayload>
      </ns1:MFTServiceInput>
    </soap:Body>
  </soap:Envelope>
```

MFT supports the SwaRef type of attachments that can be sent as part of the SOAP request by using JAVA API based clients or third party user interface clients.

Example of sFTP Reference:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soap="http://xmlns.oracle.com/fmw/mft/soap">
  <soapenv:Header>
    <soap:MFTHeader>
      <soap:TargetFilename>USD_VK_VKV_Rep_16_2.zip</soap:TargetFilename>
      <soap:ContentIdentifier>USD_VK_VKV_Rep_16_2.zip</soap:ContentIdentifier>
    </soap:MFTHeader>
  </soapenv:Header>
  <soapenv:Body>
    <soap:MFTServiceInput PayloadType="FtpRefFile">
      <soap:FTPReference>
        <soap:URL>ftp://localhost:7522/int/soap/USD_VK_VKV_Rep_16_2.zip</soap:URL>
      </soap:FTPReference>
    </soap:MFTServiceInput>
  </soapenv:Body>
</soapenv:Envelope>
```

Example of SwaRef (SOAP with Attachments) type:

```
POST /mftapp/services/transfer/mySource HTTP/1.1
Accept-Encoding: gzip, deflate
```

```
Content-Type: multipart/related; type="text/xml"; start="<rootpart@soapui.org>";
boundary="-----_Part_6_72190877.1547746700867"
SOAPAction: "http://xmlns.oracle.com/fmw/mft/soap/mftSubmit"
MIME-Version: 1.0
Content-Length: 1410
Host: localhost:7003
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

```
-----_Part_6_72190877.1547746700867
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <rootpart@soapui.org>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soap="http://xmlns.oracle.com/fmw/mft/soap">
  <soapenv:Header>
    <soap:MFTHeader>
      <soap:TargetFilename?></soap:TargetFilename>
      <soap:ContentIdentifier?></soap:ContentIdentifier>
      <soap:DeliveryOnly>
        <soap:TargetFilesize?></soap:TargetFilesize>
        <soap:SourceUser?></soap:SourceUser>
        <soap:TransferURL?></soap:TransferURL>
        <soap:SourceName?></soap:SourceName>
        <soap:TargetName?></soap:TargetName>
        <soap:TransferName?></soap:TransferName>
      </soap:DeliveryOnly>
    </soap:MFTHeader>
  </soapenv:Header>
</soapenv:Envelope>
```

```
-----_Part_6_72190877.1547746700867
Content-Type: text/xml; charset=us-ascii; name=test.xml
Content-Transfer-Encoding: 7bit
Content-ID: <test.xml>
Content-Disposition: attachment; name="test.xml"; filename="test.xml"
```

Example of binary payload:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <ns1:MFTHeader xmlns:ns1="http://
xmlns.oracle.com/fmw/mft/soap">
      <ns1:TargetFilename>Order.xml</
ns1:TargetFilename>
      <ns1:ContentIdentifier>Order.xml</
ns1:ContentIdentifier>
      <ns1:file.name>OrderIn.xml</
ns1:file.name>
    </ns1:MFTHeader>
  </soap:Header>
  <soap:Body>
    <ns1:MFTServiceInput xmlns:ns1="http://
xmlns.oracle.com/fmw/mft/soap">
      <ns1:BinaryPayload>
CQkJCTxQdXJjaGFZSU9yZGVyPg0KCQkJCQk8Y3VzdE1EPjE8L2N1c3RJRd4NCgkJCQkJPjE1EPjEwNTwvSUQ+DQoJC
QkJCTxwYXlPcHRpb24+Y3Jl
ZG10PC9wYXlPcHRpb24+DQoJCQkJCTxzaG1wQ2hvaWNlPnR3b19kYXk8L3NoaXBDbG9pY2U+DQoJCQkJCTxzdGF0d
XM+aW5pdGlhbDwvc3RhdmVz
Pg0KCQkJCQk8Y2NUeXB1PkFNRVg8L2NjVH1wZT4NCgkJCQkJPjE1EPjEwNTwvSUQ+DQoJCQkJCTxzdGF0d
2NjTnVtYmVyPg0KCQkJCQk8
aXRlbXM+DQoJCQkJCQk8aXRlbT4NCgkJCQkJCQk8cHJvZHVjdElkP1NLVTIwMDwvcHJvZHVjdElkPg0KCQkJCQkJC
Txcwcm9kdWN0TmFtZT5Wb3gg
```

```

MjAwMCAwFA8L3Byb2R1Y3ROYW11Pg0KCQkJCQkJCTxwcm1jZT4xMTAwPC9wcm1jZT4NCgkJCQkJCQk8cXVhbnRpd
Hk+MjwvcXVhbnRpdHk+DQoJ
CQkJCQk8L2l0ZW0+DQoJCQkJCQk8aXR1bT4NCgkJCQkJCQk8cHJvZHVjdElkP1NLVTEwNjwvcHJvZHVjdElkPg0KC
QkJCQkJCTxwcm9kdWN0TmFt
ZT5HaWJzb24tZGVzIFBhdWw8L3Byb2R1Y3ROYW11Pg0KCQkJCQkJCTxwcm1jZT4zMdAwPC9wcm1jZT4NCgkJCQkJC
Qk8cXVhbnRpdHk+MTwvcXVh
bnRpdHk+DQoJCQkJCQk8L2l0ZW0+DQoJCQkJCTwvaXR1bXM+DQoJCQkJPC9QdXJjaGFzZU9yZGVyPg==
        </ns1:BinaryPayload>
    </ns1:MFTServiceInput>
</soap:Body>
</soap:Envelope>

```

SOA Source Type

Using the SOA source type means transferring files from the web service interface of a SOA application.

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see *Source—SOA* in the *MFT Composer Online Help*.

Element	Description
URL	If creating a source, specifies the web service endpoint from which to transfer files. If creating a target, specifies the web service endpoint to which to transfer files. For example: <code>http://host:port/mftapp/services/transfer/url?WSDL</code> The default is localhost.
Domain Alias	If creating a source, specifies the domain from which to retrieve files. If creating a target, specifies the domain to which files are transferred.

To capture the filename or store the target filename in source, you need to pass the filename for inline payloads, as shown in the SOAP request below:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <ns1:MFTHeader xmlns:ns1="http://xmlns.oracle.com/fmw/mft/soap">
      <ns1:TargetFilename>Order.xml</ns1:TargetFilename>
      <ns1:ContentIdentifier>Order.xml</
ns1:ContentIdentifier>
      <ns1:file.name>OrderIn.xml</ns1:file.name>
    </ns1:MFTHeader>
  </soap:Header>
  <soap:Body>
    <ns1:MFTServiceInput xmlns:ns1="http://xmlns.oracle.com/fmw/mft/soap">
      <ns1:InlinePayload>
        <PurchaseOrder>
          </PurchaseOrder>
        </ns1:InlinePayload>
      </ns1:MFTServiceInput>
    </soap:Body>
  </soap:Envelope>

```

For more information about integrating Oracle Managed File Transfer with Oracle SOA Suite, see [Integrating with Oracle SOA Suite](#).

Service Bus Source Type

Using the Service Bus source type means transferring files from the web service interface of an Oracle Service Bus application.

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see *Source—Service Bus* in the *MFT Composer Online Help*.

Element	Description
URL	If creating a source, specifies the web service endpoint from which to transfer files. If creating a target, specifies the web service endpoint to which to transfer files. For example: <code>http://host:port/mftapp/services/transfer/url?WSDL</code> The default is <code>localhost</code> .
Domain Alias	If creating a source, specifies the domain from which to retrieve files. If creating a target, specifies the domain to which files are transferred.

For more information about integrating Oracle Managed File Transfer with Oracle Service Bus, see [Integrating with Oracle Service Bus](#).

B2B Source Type

Using the B2B source type means transferring files from an Oracle B2B trading partner. No settings are required if B2B is collocated. The most important settings are **Trading Partner Name**, which specifies the trading partner endpoint from which to transfer files, and **Domain Alias**, which specifies the domain from which to transfer files.

To define a trading partner in Oracle Managed File Transfer, see [Integrating with B2B](#) and [Managing Domains](#).

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see *Source—B2B* in the *MFT Composer Online Help*.

Element	Description
Domain Alias	If creating a source, specifies the domain from which to retrieve files. If creating a target, specifies the domain to which files are transferred.

To define a trading partner in Oracle Managed File Transfer, see [Integrating with B2B](#) and [Managing Domains](#).

ODI Source Type

The MFT ODI integration supports two different Source interfaces, a file Event pattern and a WebService ODI.

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see *Source—ODI* in the *MFT Composer Online Help*.

Element	Description
Binding	JCA binding types are SOAP, FTP Remote, File, and sFTP Remote.
URL	If creating a source, specifies the web service endpoint from which to transfer files. If creating a target, specifies the web service endpoint to which to transfer files. For example: <code>http://host:port/mftapp/services/transfer/url?WSDL</code> The default is <code>localhost</code> .
Domain Alias	If creating a source, specifies the domain from which to retrieve files. If creating a target, specifies the domain to which files are transferred.

To define an ODI domain in Oracle Managed File Transfer, see [Integrating with Oracle Data Integrator](#) and [Managing Domains](#).

A file Event pattern (which can be scheduled) on top of the binding endpoint type (File, FTP, sFTP) retrieves files from wherever ODI has put them. You can configure JCA source types as bindings in the ODI source along with the existing SOAP binding. You can configure the following as binding types when creating the target:

- File- Transfer file via File
- FTP Remote- Transfer file via FTP
- sFTP Remote- Transfer file via sFTP
- SOAP- Transfer file via ODI SOAP DataService

Once you have chosen one of the JCA source types, you must configure the JCA parameters as you do regular JCA sources. Once the binding type is chosen for an ODI source, you can't change the binding type but you can modify the properties for the selected binding type.

A WebService ODI source type accepts the MFT SOAP payload from the ODI application. Just like the SOAP or SOA Sources, the only required setting is URL, which specifies the web service endpoint from which to transfer files either inline or as a reference.

Storage Cloud Service Source Type

You can use Oracle Managed File Transfer to download and upload the data from an Oracle Storage Cloud Service source. Using the Storage Cloud Service source type, you can create, save, deploy, and associate the source to any transfer. Trigger and polling is available for the Oracle Storage Cloud Service endpoints. Duplicate handling is controlled by MFT. If the trigger is invoked multiple times for the same directory, a file in the directory is processed only once by MFT. MFT keeps track of already processed files.

Oracle Storage Cloud Service has an upload limitation of 5GB as the maximum file size. The actual file size can be larger than 5GB but when the large file is split into segments, the maximum segment size can be 5GB or less. In MFT target settings, a user can configure the segment size. For the outbound, a large file is split into segments that are uploaded first. A segment suffix is added to each segment, for example, `a.txt-segment-001`, `a.txt-segment-002` and so on. After uploading all the segments, the actual object manifest file (`a.txt`) is uploaded. Segment files and actual files can be uploaded in different containers. When downloading, Oracle Storage Cloud Service automatically merges the segments after the download.

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see [Source—Storage Cloud Service](#) in the *MFT Composer Online Help*.

Element	Description
User	Specifies the user who has access to the source or target. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for user names that start with \$. This is only for a leading \$. If there are other \$s in the user name, do not add more backslashes. Example: for \$john\$smith, enter the password as \john\$smith.
Password	Specifies the user password. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for passwords that start with \$. This is only for a leading \$. If there are other \$s in the password, do not add more backslashes. Example: for \$xyz\$123, enter the password as \xyz\$123.
Confirm Password	Confirms the user password.
Service URL	If creating a source, specifies the service URL from which files are transferred. If creating a target, specifies the service URL to which files are transferred. The default is localhost.
Service Name	Service Name
Container Name	Specifies the name of the container, a user-created resource, which can hold an unlimited number of objects.
Folder	If creating a source, specifies the location of files to be transferred as a directory in a file system. If creating a target, specifies the location to which files are transferred as a directory in a file system.

OCI Storage Cloud Service Source Type

Note:

You can use the OCI Storage Cloud Service source type to download data from Oracle Cloud Infrastructure and move it to any of the target types.

Note:

Before you can use the OCI Storage Cloud Service source type, you must import a private RSA key of PEM format to connect to Oracle Cloud Infrastructure. See [Importing a Key](#).

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see *Source—OCI Storage Cloud Service* in the *MFT Composer Online Help*.

Element	Description
Endpoint URL	Specifies the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage.
Region	Specifies the identifier of the Oracle Cloud Infrastructure region.

Element	Description
Namespace	Specifies the Object Storage namespace of the bucket. This is a logical entity that serves as a top-level container for all buckets and objects, allowing for control of bucket naming within your tenancy. The namespace is a unique and uneditable system-generated string assigned during account creation and applies to all regions.
Bucket Name	Specifies the name of the bucket. A bucket is a logical container for storing objects. Users or systems create buckets as needed within a region . A bucket is associated with a single <i>compartment</i> that has <i>policies</i> that determine what actions a user can perform on a bucket and on all the objects in the bucket.
Fingerprint	Specifies the fingerprint of the public API key value that you uploaded in the Oracle Cloud Infrastructure Console.
Tenant ID	Specifies the OCID of the tenancy you are using.
OCID	Specifies the OCID of the Oracle Cloud Infrastructure user account you will be using to connect to Oracle Cloud Infrastructure.
Compartment ID	Specifies the Oracle Cloud Identifier (OCID) of the compartment. Every Oracle Cloud Infrastructure resource has an Oracle-assigned unique ID called an Oracle Cloud Identifier (OCID). This is the primary building block used to organize your cloud resources. When your tenancy is provisioned, a root compartment is created for you. You can then create compartments under your root compartment to organize your resources. An Object Storage bucket can only exist in one compartment.
Private Key	Specifies the private key, which is generated using an RSA key pair of PEM format.
Passphrase	Specifies the passphrase used to encrypt the private key.
Confirm Passphrase	Confirms the passphrase used to encrypt the private key.
Test Connection	Tests the connection to OCI Storage Cloud Service with the provided parameters. Note: Test Connection returns a successful connection even if invalid values are specified for Endpoint URL or Compartment ID .

WebCenter Source Type

You can use the WebCenter source type to download data from WebCenter Content Server. You can create, save, deploy, and associate the source to any transfer.

Both the `GET_FILE` and `GET_SEARCH_RESULTS` service parameters are exposed as part of the trigger. The `GET_SEARCH_RESULTS` service in the Content Server gets the list of payloads. Using the trigger, you can also specify the specific payload that needs to be downloaded. The `GET_FILE` service is used to get the payload.

Configurable retry options via MBean are provided for the source. When all the retries are exhausted, there will be an entry in the source message table for the files that were not downloaded.

A schedule can be attached with the Content Server source. As part of a schedule, MFT calls the `GET_SEARCH_RESULTS` service with the provided `queryString` param in the source configuration.

For details about working with WebCenter Content, see *Developing with Oracle WebCenter Content*.

The following table describes settings in the Create Source dialog. For information about additional settings available after you create the source, see [Source—WebCenter](#) in the *MFT Composer Online Help*.

Element	Description
User	Specifies the user who has access to the source or target. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for user names that start with \$. This is only for a leading \$. If there are other \$s in the user name, do not add more backslashes. Example: for \$john\$smith, enter the password as \john\$smith.
Password	Specifies the user password. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for passwords that start with \$. This is only for a leading \$. If there are other \$s in the password, do not add more backslashes. Example: for \$xyz\$123, enter the password as \xyz\$123.
Confirm Password	Confirms the user password.
Protocol	Specifies the protocol for connecting to the remote server: HTTP.
Connection URL	Specifies the service endpoint URL for connecting to WebCenter applications running on remote servers. These servers can be in the same Oracle WebLogic Server domain as Oracle Managed File Transfer or in a different domain.
Selection Criteria	Enables selection of one of two modes: Query and Path.
Query Text	Text used for querying content items. The full-text search expression.

Setting Up Source Processing Actions

After you create a source, you can edit it to add processing actions such as compression, decompression, encryption or decryption, find and replace specific text, or new line character conversion to specified operating system.

You can configure processing actions for the transfer; see [Setting Up Transfer Preprocessing and Postprocessing Actions](#).

You can also create custom preprocessing and postprocessing actions; see [Processing Transfers with Custom Callouts](#).

Note:

If you add the same processing action to a source and a transfer that uses the source, the action is performed twice. For example, if you add compression to the source and the transfer, the transferred file is compressed twice.

A preprocessing decompression error occurs if a compressed file has multiple entries.

 **Note:**

If you copy a binary file to the source location using an FTP client external to Oracle Managed File Transfer, be sure to configure it for binary transfer. Otherwise the file might become corrupted. Processing actions such as compression and encryption might not work properly.

Compression and Decompression at the Source

You can compress or decompress a file prior to transfer. You can specify either action in the source configuration.

 **Note:**

Multi-file decompression is not supported for sources. It is supported only for SOAP, SOA, Service Bus, or ODI type targets as a preprocessing action or for File type targets as a postprocessing action.

The steps for this process are:

1. Click the arrow to the left of **Sources** in the left pane navigator.
The sources are listed.
2. Click the source name or right-click it and then select the Open menu item.
The source tab opens.
3. Click the arrow to the left of **Actions**.
The Actions section opens.
4. Click **add processing actions**.
The Processing Actions dialog opens.
5. Select Compress or Decompress from the All Actions drop-down list.
6. Click **Add to List**.
To remove an action from the list, click the Delete icon to the right of the action.
7. If you selected Compress, select the compression level from the Level drop-down list: Best Compression, Default Compression, or Best Speed. For more information, see the [java.util.zip](#) package, especially the `Deflater` class and the referenced specifications.
8. Click **OK**.
To cancel adding actions, click **Cancel**.
9. **Save** and optionally **Deploy** the source.

Encryption and Decryption at the Source

You can encrypt or decrypt a file prior to transfer. You can add a single encryption or decryption algorithm to the source configuration. Along with PGP algorithm, MFT supports PGP signatures. You can generate a signed and encrypted payload and validate the signature when decrypting, this can be done at the artifact level.

 **Note:**

PGP keystores must be configured and certificates must be imported before you add an encryption or decryption action.

If a payload is encrypted by a PGP tool outside of MFT using a key length or algorithm that is restricted, MFT decryption will fail. These restrictions are mostly specified at the JRE level in the `JAVA_HOME\jdk8\lib\security` directory.

The steps for this process are:

1. Click the arrow to the left of **Sources** in the left pane navigator.
The sources are listed.
2. Click the source name or right-click it and then select the Open menu item.
The source tab opens.
3. Click the arrow to the left of **Actions**.
The Actions section opens.
4. Click **add processing actions**.
The Processing Actions dialog opens.
5. Select **PGP Encryption** or **PGP Decryption** from the All Actions drop-down list.
6. Click **Add to List**.
Alternatively, to remove an action from the list, click the Delete icon to the right of the action.
7. If you selected PGP Encryption, select values from the Encryption Alias, Armored, Encryption algorithm, and Signing key alias drop-down lists:
 - Encryption Alias: the public key alias for encryption. For more information about key aliases, see [Configuring the PGP Keystore](#).
 - Armored: Binary or ASCII. Use ASCII if non-printing characters might be stripped in transit.
 - Encryption algorithm: Select from the following supported algorithms:

 **Note:**

If no algorithm is selected, global algorithm settings will apply.

- Default
- Triple-DES
- CAST5 – set as default algorithm
- Blowfish
- DES
- AES-128
- AES-192

- AES-256 – set as default algorithm, if FIPS mode is enabled
 - Twofish
 - Signing key alias: Select from the list of imported private signing keys.
If you select the signing key alias, you can override the PGP Keystore password by enabling the **Overwrite PGP Private Password** checkbox. Once enabled, you can add a Private Key Password for the artifact. The private key password overrides the existing PGP keystore's passphrase for that particular artifact (source or target).
8. If you selected PGP Decryption, select the Decryption Alias from the drop-down list. This is the private key alias for decryption. For decrypting, the signature must be imported in the PGP keystore.
- To specify a different passphrase for the PGP key, enable **Overwrite PGP Private Password** checkbox, and specify the unique passphrase. The private key password overrides the existing PGP keystore's passphrase for that particular artifact (source or target).
- For more information about key aliases, see [Configuring the PGP Keystore](#).
9. Click **OK**.
- To cancel adding actions, click **Cancel**.
10. **Save** and optionally **Deploy** the source.
- After successful transfer, you can monitor the result in the Monitor dashboard. See [Monitoring Deployed Sources, Targets, and Transfers](#) and [Transfer Reports](#).

Find and Replace at the Source

Use the Find and Replace action to replace a specified text with another text in a file prior to transfer. You can perform multiple find and replace actions on a file.

For more on find and replace actions for transfers, see [Find and Replace Preprocessing Action](#).

To Find and replace action:

1. Click the arrow to the left of **Source** in the left pane navigator.
The sources are listed.
2. Click the source name or right-click it and then select the Open menu item.
The source settings opens.
3. Click the arrow to the left of the **Actions** option.
4. Click **add processing actions**.
The Processing Actions dialog opens.
5. Select **Find and Replace** from the All Actions drop-down list.
6. Click **Add to List**. The Find and Replace Processing action is added to the Selected Actions list.
To remove an action from the list, click the Delete icon to the right of the action.
7. In the **Find** field, add the text that you want to find in the source file and in the **Replace with** field, enter the replacement text. The Find and Replacement action is case-sensitive.
You can add multiple Find and Replace text in a single Find and Replace action by clicking the Add + icon.

8. To add another replacement text, click the Add + icon.
9. You can add multiple Find and Replace processing action for each source. To add another Add and Replace action, click Find and Replace from the All Action list.
Find text is a mandatory field. If the field is blank, you will get an error.
10. Click OK.
To cancel the action, click Cancel.
After the successful transfer, the report can be seen in the Monitor Dashboard. For more information see [Monitoring Deployed Sources, Targets, and Transfers](#).

New Line Conversion at the Source

Use the New Line Conversion processing action to convert new line characters for different operating systems. The New line conversion action converts the new line character to the specified operating system specific new line character.

The steps for this process are:

1. Click the arrow to the left of Sources in the left pane navigator.
The sources are listed.
2. Click the source name or right-click it and then select the Open menu item.
The source settings opens.
3. Click the arrow to the left of the **Actions** option.
The target settings are displayed.
4. On the Target details, click **Pre-processing actions**.
The Processing Actions dialog opens.
5. Select **New line conversion** from the All Actions drop-down list and click **Add to List**.
New line conversion is displayed in Selected Actions.
6. In the **Type** field, select from the list:
 - DOS to Unix
 - Unix to DOSTo delete the New line conversion, click the Delete icon to the left of the action.
7. Click OK.
To cancel, click **Cancel**.

Run Script Processing at the Source

Use the Run Script processing action for source to execute any script or command like shell commands, perl commands or bat files on a file before delivering it to the target. You can execute external commands such as Virus Scan, external encryption file processing, add new endpoints, enable REST, notify or validate. You may run the script to modify the payload by replacing certain words or add or validate signature information during encryption and decryption.

Example of a script adding a header:

```
#bin/sh
echo "file generated by script copy"
```

```
while read line; do
echo ${line}
done
]
```

Example of a script for compression:

```
#bin/sh
gzip -c
```

Example of a error script :

```
#bin/sh
regexA="*.bak"
regexB="*.BAK"
if [[ "$fileName" == $regexA || "$fileName:" == $regexB ]]; then
    echo "Processing backup file"
else
    echo "Input file[$fileName] is not valid backup file!" 1>&2
    exit 1
fi
```

In the error script example, considering transfer is created to move backup (*.bak) files from source to target, if transfer file is not a backup (*.bak) file, transfer will fail with error "Input file[f2] is not valid backup file."

There are pre-defined variables which can be used in the script, for example `fileName` is a pre-defined variable.

To add Run Script processing action:

1. Click the arrow to the left of Sources in the left pane navigator.
The sources are listed.
2. Click the source name or right-click it and then select the Open menu item.
The source tab opens.
3. Click the arrow to the left of Actions.
The Actions section opens.
4. Click **add processing actions**.
The Processing Actions dialog opens.
5. Select **Run Script** action and click **Add to list**.
The selected action appears in the Selected Actions list.
6. Enter the following details:
 - **Command:** enter the path of the script you want to execute. The script needs to be an executable file. For example, `/home/user/echo.sh`
 - **Timeout:** Specify the timeout value, if the execution of the script takes more than specified time it will stop the execution of the script. If there is an error in the transfer,

the error is reflected in the MFT monitoring dashboard, after diagnosis it can be resubmitted.

- **Read Input Payload:** select the checkbox if you want the script (provided in Command field) to read the input payload before transfer. By default, the checkbox is selected.
 - **Use Script Generated Payload:** select the checkbox if you want to modify the existing payload with the output generated by the script. When unchecked, the script is executed without modifying the payload.
 - **New File Extension:** Specifies the extension that is added to the new file after processing is done. For example, when using compression action, you may want to modify the name to <filename>.zip, then you will set 'zip' as the new file extension.
7. Click the **Add or Update Script Variable** if you want to update, add or modify any parameter or variable in the script. There are pre-configured runtime parameters (*filename, payload directory, filesize, targetname, sourcename, useGeneratedFileFromScript*) which are updated and passed to script.
 - a. To add a variable to the script, click the Add row icon +.
 - b. Enter the **Name** of the variable and the **Value**.
 - c. If the variable value must be encrypted, select the checkbox against **Is Credential**
 - d. To delete a variable, click the Delete icon next to the Value field.
 - e. Click OK.
 8. Click OK to save the action or Cancel to cancel the action.

Archiving and Deleting Files Before Delivery

After you create a source, you can edit it to add file operations: archiving and deleting. If a file is configured to be archived, it is copied to given physical target directory. If the file is configured for deletion, it is deleted. Note that the archive or delete action applies to the target system copy of the file, not the Oracle Managed File Transfer copy of the file.

The steps for this process are:

1. Click the arrow to the left of **Sources** in the left pane navigator.
The sources are listed.
2. Click the source name or right-click it and then select the Open menu item.
The source tab opens.
3. Click the arrow to the left of **Advanced Properties**.
The Advanced properties section opens.
4. Click the **Operation** subtab.
The Operation subtab opens.
5. Select Archive, Delete, or Archive and Delete from the **Action Type** drop-down list.
6. If you selected Archive or Archive and Delete, type a path in the **Physical Target Directory** field.
7. **Save** and optionally **Deploy** the source.

Duplicating an Existing Source

You can create a new source for file transfers by copying an existing one.

The steps for this process are:

1. Duplicate a source in one of these ways:

- Select the source to copy and then the **Duplicate** icon in the left pane navigator.
- Right-click the source to copy in the left pane navigator and select the **Duplicate** command from the pop-up menu.

The Duplicate Source dialog appears.

2. Type a **Name** for the source.

The name can include letters, numbers, dashes, and underscores.

3. Type values for the required non-duplicated settings, which have blue asterisks next to them:

- **Content Folder** is required for sources of type File, FTP Remote, sFTP Remote, FTP Embedded, or sFTP Embedded.

URL is required for sources of type SOAP, SOA, Service Bus, or ODI.

Sources of type B2B have no required non-duplicated settings.

4. Click the **Create** button.

A tab for the source opens, providing additional settings you can edit. For more information about these settings, see the source type under [Source Types](#), [Setting Up Source Processing Actions](#), and [Archiving and Deleting Files Before Delivery](#).

To avoid creating a source, click **Cancel**.

5. Click the **Save** button after editing.

To undo all changes since the last save, click **Revert**.

6. Click the **Deploy** button after saving.

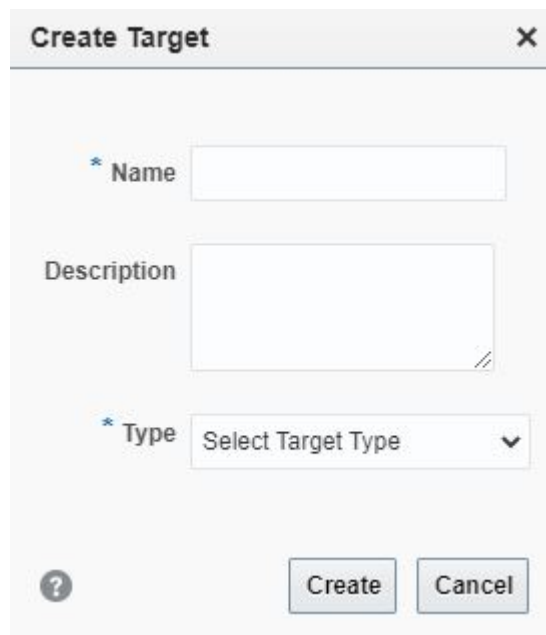
This step is optional. Deploying a transfer deploys the associated source and target automatically.

Creating a Target

You can create a target with a minimum number of settings. After you create a target, you can edit it to add more settings.

To create a target:

1. In the Designer left pane navigator, click **Targets** to open the Create Source dialog.



2. Enter a **Name** for the target.

The name can include letters, numbers, dashes, and underscores.

 **Note:**

For a SOA or SOAP target, file names must not have spaces.

3. Optionally, enter a **Description** for the target.

The description is optional.

4. Select the target **Type**.

This selection determines the other settings that appear. For more information about target types and their settings, see [Target Types](#).

5. Type a value for the target location. For most target types, this is either:

- The **Folder** setting, which specifies a file system directory. Ensure that the folder name does not exceed 60 characters.
- The **URL** setting, which specifies a web service endpoint.

The B2B target type has no target location setting in the Targets dialog. You must provide the target location after creating the target.

6. Enter values for the remaining required settings, indicated by blue asterisks.

7. Click **Create**.

A tab for the target opens, providing additional settings you can edit. For more information about these settings, see the target type under [Target Types](#) and [Moving and Renaming Files After Delivery](#).

8. After editing, click **Save**.

To undo all changes since the last save, click **Revert**.

9. (Optional) Click **Deploy**.

 **Note:**

Deploying a transfer deploys the associated source and target automatically.

If an existing target has most of the desired properties, you can duplicate it. See [Duplicating an Existing Target](#).

Target Types

Oracle Managed File Transfer provides the following target types:

- [FTP Remote Target Type](#)
- [sFTP Remote Target Type](#)
- [File Target Type](#)
- [SOAP Target Type](#)
- [SOA Target Type](#)
- [Service Bus Target Type](#)
- [B2B Target Type](#)
- [ODI Target Type](#)
- [Storage Cloud Service Target Type](#)
- [OCI Storage Cloud Service Target Type](#)
- [WebCenter Target Type](#)

Oracle Managed File Transfer does not support embedded FTP or sFTP server targets.

FTP Remote Target Type

Using the FTP Remote target type means transferring files to an FTP server outside of Oracle Managed File Transfer.

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see [Target—FTP Remote](#) in the *MFT Composer Online Help*.

Element	Description
Host Name	Specifies the host name.
Folder	If creating a source, specifies the location of files to be transferred as a directory in a file system. If creating a target, specifies the location to which files are transferred as a directory in a file system.
User	Specifies the user who has access to the source or target. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for user names that start with \$. This is only for a leading \$. If there are other \$s in the user name, do not add more backslashes. Example: for \$john\$smith, enter the password as \<\$john\$smith.

Element	Description
Password	Specifies the user password. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for passwords that start with \$. This is only for a leading \$. If there are other \$s in the password, do not add more backslashes. Example: for \$xyz\$123, enter the password as \<\$xyz\$123.
Confirm Password	Confirms the user password.
Control Port	Specifies the port for the source or target.
SSL	Specifies the use of SSL if checked. This is optional.
Implicit SSL	Specifies the use of implicit SSL if checked. This is optional.

FTP Remote Target settings for MVS Transfers

When creating a FTP Remote target type for MVS Mainframe systems, you need to select MVS as List Parser Key in advance properties.

For MVS FTP response formats, MVS can be configured to use HFS (Unix style) response or MVS native response formats. If the target type uses MVS HFS (Unix style), you can configure the advanced Properties as you do with any other Unix system. For example, using "/" as the path separator. However, if the MVS system uses a MVS native response format only, then you must configure the FTP Remote target type with following mandatory properties:

- The "Content Folder" field is not used, however it is a mandatory field. You must enter some text as a placeholder and cannot be left as blank filed. For example, Content Folder = "FOLDER."
- The FTP Path Separator field must be left blank/empty. For example, FTP Path Separator = ""
- The "File Naming Convention" field specifies the absolute path to the file. You must include the filename in the path, and enclose it in single quotes. This field must be in the format, File Naming Convention="MFTOUT.MFT%SEQ%.CSV". For example, 'QA.TEST.FILE '.
- Change the Default Date Format. For example, yyyy/mm/dd:
- Absolute Path Begin = ""
- List Parser Key = MVS

You can configure the advance properties as shown in the figure listed below:

sFTP Remote Target Type

Using the sFTP Remote target type means transferring files to an sFTP server outside of Oracle Managed File Transfer.

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see Target—sFTP Remote in the *MFT Composer Online Help*.

Element	Description
Host Name	Specifies the host name.
Folder	If creating a source, specifies the location of files to be transferred as a directory in a file system. If creating a target, specifies the location to which files are transferred as a directory in a file system.
User	Specifies the user who has access to the source or target. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for user names that start with \$. This is only for a leading \$. If there are other \$s in the user name, do not add more backslashes. Example: for \$john\$smith, enter the password as \john\$smith.
Password	Specifies the user password. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for passwords that start with \$. This is only for a leading \$. If there are other \$s in the password, do not add more backslashes. Example: for \$xyz\$123, enter the password as \xyz\$123.
Confirm Password	Confirms the user password.
Control Port	Specifies the port for the source or target.
Authentication Type	Specifies the type of user authentication: Password or Public Key.

File Target Type

Using the File target type means transferring files to the local file system or to a network-attached system. The only required setting is **Folder**, which specifies the directory to which to transfer files. This directory must be accessible from Oracle Managed File Transfer.

Oracle Managed File Transfer uses the same file adapter used by Oracle SOA Suite.

For information about additional settings available after you create the target, see Target—File in the *MFT Composer Online Help*.

SOAP Target Type

Using a SOAP web service type means transferring files to a web service. The only required setting is **URL**, which specifies the web service endpoint to which to transfer files.

For more information about integrating with Oracle Managed File Transfer as a web service, see [Integrating with Web Services](#).

For information about additional settings available after you create the target, see Target—SOAP in the *MFT Composer Online Help*.

SOA Target Type

Using the SOA target type means transferring files to the web service interface of a SOA application.

For more information about integrating Oracle Managed File Transfer with Oracle SOA Suite, see [Integrating with Oracle SOA Suite](#).

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see Target—SOA in the *MFT Composer Online Help*.

Element	Description
URL	If creating a source, specifies the web service endpoint from which to transfer files. If creating a target, specifies the web service endpoint to which to transfer files. For example: <code>http://host:port/mftapp/services/transfer/url?WSDL</code> The default is <code>localhost</code> .
Domain Alias	If creating a source, specifies the domain from which to retrieve files. If creating a target, specifies the domain to which files are transferred.

Service Bus Target Type

Using the Service Bus target type means transferring files to the web service interface of an Oracle Service Bus application. The only required setting is **URL**, which specifies the web service endpoint to which to transfer files.

For more information about integrating Oracle Managed File Transfer with Oracle Service Bus, see [Integrating with Oracle Service Bus](#).

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see Target—Service Bus in the *MFT Composer Online Help*.

Element	Description
URL	If creating a source, specifies the web service endpoint from which to transfer files. If creating a target, specifies the web service endpoint to which to transfer files. For example: <code>http://host:port/mftapp/services/transfer/url?WSDL</code> The default is <code>localhost</code> .
Domain Alias	If creating a source, specifies the domain from which to retrieve files. If creating a target, specifies the domain to which files are transferred.

B2B Target Type

Using the B2B target type means transferring files to an Oracle B2B trading partner. No settings are required if B2B is collocated. The most important settings are **Trading Partner Name**, which specifies the endpoint to which to transfer files, and **Domain Alias**, which specifies the domain to which to transfer files.

To define a trading partner in Oracle Managed File Transfer, see [Integrating with B2B](#) and [Managing Domains](#).

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see Target—B2B in the *MFT Composer Online Help*.

Element	Description
Domain Alias	If creating a source, specifies the domain from which to retrieve files. If creating a target, specifies the domain to which files are transferred.

ODI Target Type

The MFT ODI integration supports two different Target interfaces, a file deliver and notify pattern and a WebService interface.

Using a file deliver and notify pattern, MFT delivers the file to one of the binding types listed below then invokes the `OdilInvokeWebService` tool to process the file. Neither an inline file nor a reference is provided as part of the `OdilInvokeWebService` invocation. After selecting one of the binding types, you must configure the required parameters for the targets. Once a binding type is selected for an ODI target, you can't change the binding type, but you can continue modifying the parameters of the current target binding. For the File, FTP Remote and sFTP Remote bindings, optionally, an ODI Invoke Post Processing Action can be configured to invoke an ODI scenario to then process the file.

- File- Transfer file via File
- FTP Remote- Transfer file via FTP
- sFTP Remote- Transfer file via sFTP

For more information about the `OdilInvokeWebService`, see [ODIInvoke Post-Processing Actions](#)

Using the WebService interface, MFT invokes the ODI SOAP Data Services interface. This is typically used to update a single table and not suited for file delivery.

To define an ODI domain in Oracle Managed File Transfer, see [Integrating with Oracle Data Integrator](#) and [Managing Domains](#).

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see Target—ODI in the *MFT Composer Online Help*.

Element	Description
Binding	JCA binding types are SOAP, FTP Remote, File, and sFTP Remote.
URL	If creating a source, specifies the web service endpoint from which to transfer files. If creating a target, specifies the web service endpoint to which to transfer files. For example: <code>http://host:port/mftapp/services/transfer/url?WSDL</code> The default is <code>localhost</code> .
Domain Alias	If creating a source, specifies the domain from which to retrieve files. If creating a target, specifies the domain to which files are transferred.

Storage Cloud Service Target Type

Using the Storage Cloud Service target type you can create, save, deploy, and associate source to any transfer. Scheduling is supported for the Storage Cloud Service target.

Storage Cloud Service has a limitation of 5GB as the maximum file size that can be uploaded to Storage Cloud Service. MFT segments the files for you so you can easily upload a file larger than 5GB.

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see Target—Storage Cloud Service in the *MFT Composer Online Help*.

Element	Description
User	Specifies the user who has access to the source or target. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for user names that start with \$. This is only for a leading \$. If there are other \$s in the user name, do not add more backslashes. Example: for \$john\$smith, enter the password as \ \$john\$smith.
Password	Specifies the user password. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for passwords that start with \$. This is only for a leading \$. If there are other \$s in the password, do not add more backslashes. Example: for \$xyz\$123, enter the password as \ \$xyz\$123.
Confirm Password	Confirms the user password.
Service URL	If creating a source, specifies the service URL from which files are transferred. If creating a target, specifies the service URL to which files are transferred. The default is <code>localhost</code> .
Service Name	Specifies the service name of the target.
Container Name	Specifies the name of the container, a user-created resource, which can hold an unlimited number of objects.
Folder	If creating a source, specifies the location of files to be transferred as a directory in a file system. If creating a target, specifies the location to which files are transferred as a directory in a file system.

OCI Storage Cloud Service Target Type

 **Note:**

You can use the OCI Storage Cloud Service target type to upload data to Oracle Cloud Infrastructure from any of the source types.

 **Note:**

Before you can use the OCI Storage Cloud Service target type, you must import a private RSA key of PEM format to connect to Oracle Cloud Infrastructure. See [Importing a Key](#).

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see Target—OCI Storage Cloud Service in the *MFT Composer Online Help*.

Element	Description
Endpoint URL	Specifies the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage.
Region	Specifies the identifier of the Oracle Cloud Infrastructure region.
Namespace	Specifies the Object Storage namespace of the bucket. This is a logical entity that serves as a top-level container for all buckets and objects, allowing for control of bucket naming within your tenancy. The namespace is a unique and uneditable system-generated string assigned during account creation and applies to all regions.
Bucket Name	Specifies the name of the bucket. A bucket is a logical container for storing objects. Users or systems create buckets as needed within a region . A bucket is associated with a single <i>compartment</i> that has <i>policies</i> that determine what actions a user can perform on a bucket and on all the objects in the bucket.
Fingerprint	Specifies the fingerprint of the public API key value that you uploaded in the Oracle Cloud Infrastructure Console.
Tenant ID	Specifies the OCID of the tenancy you are using.
OCID	Specifies the OCID of the Oracle Cloud Infrastructure user account you will be using to connect to Oracle Cloud Infrastructure.
Compartment ID	Compartment ID
Private Key	Specifies the private key, which is generated using an RSA key pair of PEM format.
Passphrase	Specifies the passphrase used to encrypt the private key.
Confirm Passphrase	Confirms the passphrase used to encrypt the private key.
Test Connection	Tests the connection to OCI Storage Cloud Service with the provided parameters. Note: Test Connection returns a successful connection even if invalid values are specified for Endpoint URL or Compartment ID .

WebCenter Target Type

You can use the WebCenter target type to upload data to Webcenter Content Server. You can create, save, deploy, and associate the target to any transfer.

The following table describes settings in the Create Target dialog. For information about additional settings available after you create the target, see *Target—WebCenter* in the *MFT Composer Online Help*.

Element	Description
User	Specifies the user who has access to the source or target. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for user names that start with \$. This is only for a leading \$. If there are other \$s in the user name, do not add more backslashes. Example: for \$john\$smith, enter the password as \john\$smith.
Password	Specifies the user password. MFT treats properties beginning with \$ as parameters. Add a backslash (\) before the \$ for passwords that start with \$. This is only for a leading \$. If there are other \$s in the password, do not add more backslashes. Example: for \$xyz\$123, enter the password as \xyz\$123.
Confirm Password	Confirms the user password.
Protocol	Specifies the protocol for connecting to the remote server: HTTP.
Connection URL	Specifies the service endpoint URL for connecting to WebCenter applications running on remote servers. These servers can be in the same Oracle WebLogic Server domain as Oracle Managed File Transfer or in a different domain.

Moving and Renaming Files After Delivery

After you create a File, FTP Remote, or SFTP Remote target, you can edit it to add file operations: moving and renaming.

The steps for this process are:

1. Click the arrow to the left of **Targets** in the left pane navigator.
The targets are listed.
2. Click the target name or right-click it and then select the Open menu item.
The target tab opens.
3. Click the arrow to the left of **Advanced Properties**.
The Advanced properties section opens.
4. Click the **Operation** subtab.
The Operation subtab opens.
5. Select Move, Rename, or Move and Rename from the **Action Type** drop-down list.
6. If you selected Move or Move and Rename, type a path in the **Physical Target Directory** field.
7. If you selected Rename or Move and Rename, type a file name pattern in the **File Naming Convention** field.

You can use variables in the file name such as `%yyMMddHHmmssSSS%` for the timestamp or `%SEQ%` for an incrementing integer. For example, `File%SEQ%.txt` numbers files `File1.txt`, `File2.txt`, and so on. See "Specifying the Outbound File Naming Convention" in *Understanding Technology Adapters* for more information.

8. **Save** and optionally **Deploy** the target.

Duplicating an Existing Target

You can create a new target for file transfers by copying an existing one.

The steps for this process are:

1. Duplicate a target in one of these ways:
 - Select the target to copy and then the **Duplicate** icon in the left pane navigator.
 - Right-click the target to copy in the left pane navigator and select the **Duplicate** command from the pop-up menu.

The Duplicate Target dialog appears.

2. Type a **Name** for the target.

The name can include letters, numbers, dashes, and underscores.

3. Click the **Create** button.

A tab for the target opens, providing additional settings you can edit. For more information about these settings, see the target type under [Target Types](#) and [Moving and Renaming Files After Delivery](#).

To avoid creating a target, click **Cancel**.

4. Click the **Save** button after editing.

To undo all changes since the last save, click **Revert**.

5. Click the **Deploy** button after saving.

This step is optional. Deploying a transfer deploys the associated source and target automatically.

Setting Up Schedules

You can schedule file deliveries so they occur only at specific times or time ranges. If no schedule is configured, the file is delivered as soon as it is processed by Oracle Managed File Transfer. You can configure a source schedule as part of source configuration or a target schedule as part of transfer configuration.

If a schedule is defined for a listening source, the file is picked only when the schedule expires. For a non-listening source, the file is picked as soon as it arrives at the source location but remains with a status of Scheduled. When the schedule expires, the file is processed further and delivered. All transfers that reference the source happen only when the source schedule expires.

If a schedule is defined for a target, the file is delivered only when the schedule expires. Before that it remains at the source location with a status of Scheduled. Targets referenced by the same transfer do not share schedules.

 **Note:**

Before adding a schedule, test the transfer without one to ensure that it works properly. See [Deploying and Testing Transfers](#).

 **Note:**

Oracle Managed File Transfer interacts with Oracle Enterprise Scheduler Service through the `OracleSystemUser`. Do not delete this user. If you do, clicking **add schedule** will result in an `OracleSystemUser does not exist` message, and **Schedule Details** may be blank in monitoring reports. For more information about users, see [Configuring Users](#).

The steps for this process are:

1. Click the arrow to the left of **Sources** or **Transfers** in the left pane navigator.
The sources or transfers are listed.
2. Click the source or transfer name or right-click it and then select the Open menu item.
The source or transfer tab opens.
3. If you don't see the **add schedule** option, you can display it. In the source tab, click the arrow to the left of Source Schedule. In the transfer tab, click the arrow to the left of the target.
4. Click **add schedule**.
The Scheduler dialog opens.
5. Type a **Name** for the schedule.
6. Select a value from the **Frequency** drop-down list: Once, Hourly/Minute, Daily, Weekly, Monthly, Yearly, or Custom.
 - If you selected Once, enter a date and time in the **Start Date** field.
 - If you selected Hourly/Minute, specify the interval in hours and minutes. Enter a **Start Date** and optionally an **End Date**.
 - If you selected Daily, specify the interval in days. Enter a **Start Date** and optionally an **End Date**.
 - If you selected Weekly, specify the interval in weeks. Enter a **Start Date** and optionally an **End Date**.
 - If you selected Monthly, select a **Repeat** option:
 - **By day:** Select a **Week of the Month** and a **Day of the Week**. For the **Week of the Month**, you can select **Last**.
 - **By date:** Select a day of the month or **Last day of month**.Enter a **Start Date** and optionally an **End Date**.
 - If you selected Yearly, select a **Month** and a **Repeat** option:
 - **By day:** Select a **Week of the Month** and a **Day of the Week**. For the **Week of the Month**, you can select **Last**.

- **By date:** Select a day of the month or **Last day of month**.

Enter a **Start Date** and optionally an **End Date**.

Click the Date/Time icon to select a date and time instead of typing them or to select a different time zone. Click **Customize Times** to edit individual delivery times.

7. If you selected Custom from the **Frequency** drop-down list or clicked **Customize Times**, the Scheduler dialog expands and displays a table of times.

To add a time, click **Add Time**, specify a date and time in the Add Time dialog, and click **OK**. Repeat for each time you want to add.

To remove a time, click **Remove Time**. To restore a removed time, click **Add Back**.

If you clicked **Customize Times**, you can cancel adding custom times by clicking **Change Frequency**.

8. To specify a time range in which file deliveries can occur, check **Use Duration**. Specify the duration in hours and minutes. The duration must be more than the **Frequency**.

The duration is the range of time in which transfers occur. For example, if the Frequency is Weekly, the Start Date is a Monday at noon, and the Duration is one hour, then polling or transfers occur only on Mondays between noon and 1 pm.

9. Click **OK**.

To cancel adding a schedule, click **Cancel**.

10. **Save** and **Deploy** the source or transfer.

Schedules with Polling Frequency and Minimum Age

The minimum age applies to the listening source types listed below:

- Remote sFTP
- Remote FTP
- File
- Storage Cloud Service
- Webcenter

If a schedule is defined for any of these source types, polling frequency and minimum age applies only to the schedule duration. If the schedule ends before the expiration of polling frequency, then the listening source types will not be polled.

Upon each polling occurrence or schedule expiration, MFT downloads only the files that have a last modified time to a value that is larger than minimum age.

Note:

When using the same file name for multiple transfers, some times the release of the lock is delayed and cause issues with the file processing, as the filename is the key on the lock table. The files are not picked after every polling interval due to lock. To resolve this issue, add a MFT MBean property `releaseLockCycle` and set its value to 1 and redeploy the source. This fix applies to File, FTP remote, SFTP remote sources.

Setting Up Events

The Events service enables you to trigger a file transfer on demand. You can trigger file transfers for sources such as File, FTP Remote, sFTP Remote, WebCenter, and Oracle Storage Cloud Service. If the Events service is enabled for a source, it triggers an immediate retrieval of files on external invocation. By default, the trigger service is protected by username token policy.

You can invoke Events using the following methods:

- MFT console
- WLST commands: refer `triggerEvent` in [MFT WLST Command Summary](#)
- REST Service: refer REST API for Oracle Managed File Transfer
- SOAP Service: refer [Trigger Events using REST and SOAP Services](#)

Enabling Events for a source

The Events service can be enabled by choosing the Source Schedule type as Event. To secure the service, select the **Security** checkbox.

To enable Events using the MFT console:

1. Click the arrow to the left of **Sources** in the left pane navigator. The sources are listed.
2. Click the source name or right-click it and then select the **Open** menu item.
(Optional) The source or transfer tab opens and source details are displayed.
3. Click the **Source Schedule** option. Polling and Events options are visible.
4. Enable the Events for the selected source by selecting the **Events** checkbox.

Events cannot be enabled if Polling is enabled. If you need Polling and Events service to be enabled, add a schedule. Else, disable Polling and enable Events.

If the same source is configured for Polling and Events, then if polling is in process, Events request is not accepted. If an Event is in process, then Polling is paused until the Event is completed.

5. Click the **Security** checkbox to secure the events. By securing the events, only Administrators and configured MFT users can trigger an event. For information on configuring users and roles, see [Configuring Users](#).
6. Click **Save** to save the changes.

To invoke the Events service:

1. From the **Monitoring** tab, click the arrow to the left of **Source Instances** in the left pane navigator.
2. Click the source name or right click it and then select the **Open** menu item.
3. On the **Instances** tab, select the source and click **Invoke Event**.

After you trigger an event, you get an Event session ID which is used to view the Event details and status in the Web Service client.

Trigger Events using REST and SOAP Services

Once you have set up Events, you can trigger it on demand using REST or SOAP operations.

For SOAP Services, URL is `http://<host>:<port>/mftapp/services/MFTEventService`

For REST services, refer REST API for Oracle Managed File Transfer

Supported SOAP Operations

The supported SOAP operations for trigger Events are:

- `submitEvent` - To submit an Event which will trigger the transfer, you need to supply the source name as a mandatory parameter along with additional optional parameters. On successful invoke of Event, an `eventSessionId` is returned which can be used to query the Event status and instances.

Supported parameters for JCA sources: `PatternType` - wildcard or regular expression, `IncludePattern` - any wildcard or regular expression filter string, `ExcludePattern` - any wildcard or regular expression filter string. `PatternType` is a mandatory input parameter if you are passing pattern. These properties override values defined at the source level.

Example: `curl -s -u username:password -H Content-Type:application/json -X POST -d @ '{"sourceName": "Wile SFTP Remote Source", "properties": {"entry": [{"key": "PatternType", "value": "wildcard"}, {"key": "ExcludePattern", "value": "*sh"}]}}' http://localhost:7003/mftapp/rest/v1/events`

Supported parameters for Oracle Cloud Storage sources: `PayloadKey`, `Path`, `Delimiter`

Supported parameters for WebCenter Sources: `DocName`, `DocId`, `RevisionSelectionMethod`, `Rendition`, `Querytext`, `QueryFormat`, `SecurityGroup`, `DocumentAccount`

- `getInstanceDetails` - Get the details of instances created by the Event. It requires `eventSessionId` as mandatory input parameter. The response contains the details of each instance (one instance per file). By default only minimal info is provided, to get the full details you need to set the `inDetail` attribute to true.
- `getEventStatus` - To get the overall Event status about counts of how many files are processed successfully or failed or in progress along with the Event status (for example Done or errored). It requires `eventSessionId` as mandatory input parameter. The response contains the count of instances in different states and status of the Event.

All the operations of the Event SOAP service are protected by user name token policy, to invoke MFT Event Service in SOAP, you must provide the user name token client policy.

Setting Up Priorities

You can define priorities for transfers so that the messages of High priority are processed first.

You can control the order message processing and transfer of the payload based on the priority of the associated transfer where messages of *HIGH* priority are processed first, followed by *MEDIUM*, then *LOW*. If the priority of more than one transfer is the same, then the messages are picked and processed in order of submission. Priority is an attribute added to the MFT meta model and the priority defined on the Transfer page is persisted in the MDS.

The message priority is shown in the reports for source/transfer/target modules. The priority at source is inherited from the transfer. If there is more than one transfer then the highest priority of all transfers is considered for the source level message processing and is shown in the reports.

You can view the priority in artifact specific instances in the Monitoring dashboard. The Monitoring dashboard also provides a filter so that you can search the messages based on priority.

Deploying and Testing Transfers

After you create a transfer and its associated source and targets, you deploy the transfer to activate it, then test it to ensure it works as designed.



Note:

Before adding a schedule, test the transfer without one to ensure that it works properly. See [Setting Up Schedules](#).

Deploying a Source, Target, or Transfer

Every artifact tab has a **Deploy** button. Before a transfer can deliver files, it must be deployed. You can deploy sources and targets separately to make them available for use in multiple transfers.

The deployment process has three steps:

1. The deployment user interface displays the list of files to be deployed.
2. The files are validated.
3. If the validation is successful, the artifact is deployed.

Deploying a transfer for the first time automatically deploys the associated source and targets if they have been saved but not deployed. However, after the initial deployment, each artifact must be separately redeployed after any modifications.

Oracle Managed File Transfer maintains the version of the artifact. When an artifact is deployed, the current version of the artifact is deployed. The Oracle Managed File Transfer runtime engine operates only on the deployed version.

You can monitor, disable, and undeploy artifacts that have been deployed; see [Monitoring Deployed Sources, Targets, and Transfers](#).

How to Tell If a Transfer Is Successful

To test a deployed transfer, copy to the source location a test file of the type the transfer is designed to deliver.

If you applied a content filter, you can also verify that a file of the wrong type is not transferred. See [Setting Up Content Filters](#).



Note:

If you copy a binary file to the source location using an FTP client external to Oracle Managed File Transfer, be sure to configure it for binary transfer. Otherwise the file might become corrupted. Processing actions such as compression and encryption might not work properly.

Locating Received Files

You can verify that the transfer worked by verifying that the test file arrived at the target location.

If you applied preprocessing or postprocessing actions such as compression or encryption, you can examine the delivered file to verify that these actions occurred. See [Setting Up Source Processing Actions](#) and [Setting Up Transfer Preprocessing and Postprocessing Actions](#).

You can also verify that actions such as moving and renaming occurred. See [Archiving and Deleting Files Before Delivery](#) and [Moving and Renaming Files After Delivery](#).

Watching Active Deliveries

If your test file is large, you can watch its progress on the Dashboard tab of the Monitoring page. See [Monitoring Active Deliveries](#).

Importing and Exporting Transfers

Exporting a transfer saves the transfer configuration and its associated source and target configurations to a ZIP file. Sources and targets cannot be exported separately.

To export a transfer, open the transfer and click the Export button. You can download the transfer with or without a config plan.

You can import a transfer you have previously exported. The steps for this process are:

1. Open the Import/Export tab on the Administration page.
2. Click **Browse**.

An operating system file uploading dialog box opens.

3. Select the directory from which to upload the file.
4. Select the ZIP file to upload.
5. Click **Open**.

The full path to the file appears in the Import text box next to the Browse button.

6. Click **Import**.

This overwrites a transfer artifact with the same name, and overwrites any associated source and targets with the same names.

3

Processing Transfers with Custom Callouts

Learn how to create and use custom callouts for file transfer preprocessing and postprocessing in Oracle Managed File Transfer (MFT).

To create callouts, you must be familiar with Java and XML code and with creating transfers as described in [Designing Artifacts: Transfers, Sources, and Targets](#).

This chapter includes the following sections:

- [Understanding Custom Callouts](#)
- [Creating a Custom Callout: High-Level Steps](#)
- [Creating the Code](#)
- [Creating the Callout Definition File](#)
- [Locating the Callout Directory](#)
- [Running the createCallouts Command](#)
- [Testing the Callout](#)
- [Reference Files](#)
- [Validating Checksum on Custom Callout](#)

Understanding Custom Callouts

Oracle Managed File Transfer provides built-in compression, decompression, encryption, and decryption actions for transfer preprocessing and postprocessing actions.

You can create new preprocessing and postprocessing actions, which are called *custom callouts*. For detailed information, see [Setting Up Transfer Preprocessing and Postprocessing Actions](#) and [Setting Up Source Processing Actions](#).

Examples of custom callout uses are as follows:

- Adding copyright headers or footers
- Validating content
- Notifying a separate application in case of delivery success or failure
- Invoking a separate application in case of delivery success or failure
- Adding custom headers or properties, which are persisted in the MFT database
- Adding an alternative encryption technique

This chapter uses a Newline Conversion callout example. For additional callout samples, see <http://www.oracle.com/technetwork/middleware/mft/index.html>.

Custom callouts can be associated with either the source or the target. The sequence of processing action execution during a transfer is as follows:

1. Source preprocessing actions
2. Target preprocessing actions

3. Payload delivery
4. Target postprocessing actions

 **Note:**

In the MFT console, target processing actions are configured at the transfer. In [Designing Artifacts: Transfers, Sources, and Targets](#), these actions are referred to as *transfer* processing actions. In this chapter, these actions are referred to as *target* processing actions.

 **Note:**

Postprocessing occurs after file delivery. Therefore, the Active Deliveries and File Finder views in the Dashboard tab on the Monitoring page show different statuses if file delivery succeeds but postprocessing fails. Specifically, the Active Deliveries view displays a Completed status but the File Finder view displays a Failed status.

[Table 3-1](#) summarizes the types of callouts you can create.

Table 3-1 Callout Types

Callout Type	Interface Name	Payload Modification	Header and Property Modification
Preprocessing	PreCalloutPlugin	Allowed	Allowed
Postprocessing	PostCalloutPlugin	Not Allowed	Allowed
Postprocessing	PostDeliveryFailureCalloutPlugin	Not Allowed	

Creating a Custom Callout: High-Level Steps

Learn how to create custom callouts.

The high-level steps for creating a custom callout are:

1. Create the Java source code based on the appropriate interface and package it in a JAR file. See [Creating the Code](#).
2. Create a callout definition XML file based on the callout schema. See [Creating the Callout Definition File](#).
3. Copy the JAR file and the XML file into the callout directory. See [Locating the Callout Directory](#).
4. Run the createCallouts WLST command to configure the callout in MFT. See [Running the createCallouts Command](#).
5. Test the callout by using it in a file transfer. See [Testing the Callout](#).

The following sections describe these steps in detail using a Newline Conversion callout example. A newline, also known as a line break or end-of-line (EOL) marker, is a special character or sequence of characters signifying the end of a line of text. Unix operating systems

use different newline characters than DOS (and Windows) operating systems. Using this callout, you can convert newline characters from DOS to Unix or from Unix to DOS.

Creating the Code

One of the steps to create a custom callout is to create the Java code.

This section includes the following topics:

- [Java Code Requirements and Tips](#)
- [Java Code for the Newline Conversion Example](#)

Java Code Requirements and Tips

The Java code defines the callout actions. It must meet these minimum requirements:

- It must import the `java.io.IOException`, `java.io.InputStream`, `java.io.OutputStream`, and `java.util.Map` classes.
- It must import the `PluginOutput` and `PluginContext` classes in the `com.oracle.tip.mft.engine.processor.plugin` package.
- It must implement the `PreCalloutPlugin` or `PostCalloutPlugin` interface in the `com.oracle.callout` package.

See [PreCalloutPlugin Interface](#) or [PostCalloutPlugin Interface](#) for the contents of the interface.

- It must be packaged in a JAR file.

You can include multiple callouts in the same JAR file or package them in separate JAR files.

To make a callout set header variables, use code like this:

```
context.getCustomPropertyMap().put("name", "value");
context.getTransportHeaderMap().put("name", "value");
```

To make a callout change the payload file name, use code like this:

```
PluginOutput pOutput = new PluginOutput();
    pOutput.setNewFileName("abc.xyz");
    return pOutput;
```

To make a callout perform different actions for source preprocessing, source postprocessing, target preprocessing, and target postprocessing, use code like this:

```
if(message instanceof oracle.tip.mft.bean.SourceMessage){
    // in case of source pre and post processing
} else if(message instanceof oracle.tip.mft.bean.Instance){
    // in case of target pre
} else if(message instanceof oracle.tip.mft.bean.TargetMessage ){
    // in case of target post
}
```

To compile the code, use a command like this, with the core MFT JAR file in the classpath:

```
javac -classpath $MW_HOME/mft/modules/oracle.mft/core.jar <class>
```

To create the JAR file, use a command like this:

```
jar cf newlineConversion.jar
```

Java Code for the Newline Conversion Example

The Java class for the Newline Conversion example performs these actions:

1. Specifies that the callout changes the payload.
2. Accepts the `Type` parameter value and determines whether it is `Dos2Unix` or `Unix2Dos`.
3. Returns null if no `Type` value is provided.
4. Uses the `doLineConversion` method to rewrite each line in the payload with the chosen newline characters.
5. Returns a copy of the payload file with the newlines changed.

Example - Newline Conversion Code shows the Java class code for the Newline Conversion example.

```
package com.oracle.callout.sample;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.util.Map;

import oracle.tip.mft.engine.processor.plugin.PluginContext;
import oracle.tip.mft.engine.processor.plugin.PluginOutput;
import oracle.tip.mft.engine.processor.plugin.PreCalloutPlugin;

public class NewlineConversion implements PreCalloutPlugin {

    @Override
    public boolean isPayloadChangeRequired(PluginContext context,
        Map<String, String> calloutParams) {
        return true;
    }

    @Override
    public PluginOutput process(PluginContext context, InputStream input,
        OutputStream out, Map<String, String> calloutParams) {
        String type = calloutParams.get("Type");
        boolean isDos = false;
        if ("Unix2Dos".equals(type)) {
            isDos = true;
        }
        doLineConversion(input, out, isDos);
        return new PluginOutput();
    }

    @Override
    public PluginOutput process(PluginContext context, InputStream input,
        Map<String, String> calloutParams) {
        return null;
    }
}
```

```
public static final String DOS_NEW_LINE = "\r\n";
public static final String UNIX_NEW_LINE = "\n";

private void doLineConversion(InputStream in, OutputStream out,
    boolean isDos) {
    String newLineChar = null;
    if (isDos) {
        newLineChar = DOS_NEW_LINE;
    } else {
        newLineChar = UNIX_NEW_LINE;
    }
    BufferedReader bufferIn = null;
    BufferedWriter bufferOut = null;
    try {
        DataInputStream dataIn = new DataInputStream(in);
        bufferIn = new BufferedReader(new InputStreamReader(dataIn));
        DataOutputStream dataOut = new DataOutputStream(out);
        bufferOut = new BufferedWriter(new OutputStreamWriter(dataOut));
        // For each line in the un-normalized file
        String line;
        while ((line = bufferIn.readLine()) != null) {
            // Write the original line plus the operating-system dependent
            // newline
            bufferOut.write(line);
            bufferOut.write(newLineChar);
        }
    } catch (Exception e) {
        throw new RuntimeException(e);
    } finally {
        try {
            bufferIn.close();
            bufferOut.close();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Creating the Callout Definition File

The definition file specifies the callout structure in the MFT configuration. The file must meet certain requirements to be valid, for example, the xml format, attributes values, input parameters values must be correct.

The requirements to create the callout definition file are:

- It must conform to the XML format specified in the `callout.xsd` schema file. See [Callout Definition Schema](#) for the contents of the schema file.

You can include multiple callouts in the same definition file or package them in separate definition files.

- The `timeout` attribute of the `Callout` element must have a value (in seconds) to prevent an insufficiently debugged callout with an infinite loop from running too long.
- The `implementationClass` attribute of the `Callout` element must reference the Java class name, including the package name.
- The `libraryName` attribute of the `Callout` element must reference the JAR file name.

- If the Java class has input parameters, corresponding `parameter` elements must be specified.

Newline Conversion Callout Definition File shows the callout definition file for the Newline Conversion example.

```
<?xml version="1.0" encoding="UTF-8"?>
<mft:Callouts xmlns:mft="http://xmlns.oracle.com/mft"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/mft callout.xsd ">
  <mft:Callout description="New line conversion"
    helpText="New line conversion"
    groupName="Source-pre,Target-pre" timeout="300"
    implementationClass="com.oracle.callout.sample.NewlineConversion"
    libraryName="newlineConversion.jar" name="New line conversion">
    <mft:Parameter description="Type" mandatory="true" helpText="Type"
      name="Type" listValue="Dos2Unix,Unix2Dos"
      parameterType="string" defaultValue="Dos2Unix"/>
  </mft:Callout>
</mft:Callouts>
```

Locating the Callout Directory

After you have created the JAR file and the definition file, you must copy them into the MFT callout directory. The default location of the callout directory can be verified in the MFT configuration.

The default location of this directory is as follows:

```
FMW_HOME/user_projects/domains/soainfra/mft/callouts
```

To verify the location, you can look at the MFT configuration.

The steps for this process are:

1. Go to the Administration page.
2. In the left navigation pane, click **Server Properties**.
3. Look at the **Callout Directory** setting near the top of the page.

Running the createCallouts Command

After you have copied the JAR file and the definition file into the MFT callout directory, use the `createCallouts` command to upload the callout definition to the MFT configuration.

Follow the below steps to run the `createCallout` command:

1. Start WLST using the steps described in [Running WLST Commands](#).
2. Specify the full path to the callout definition file using the `createCallouts` command:

```
crtCalls('/home/user_projects/domains/soainfra/mft/callouts/Newline.xml')
```

The example uses the `crtCalls` abbreviation.

3. Verify that the callout has been added using the `listCallouts` command:

```
listCallouts()
```

4. Exit WLST using the steps described in [Running WLST Commands](#).

For more information about MFT WLST commands, see [MFT WLST Command Summary](#) and MFT Custom WLST Commands in *WLST Command Reference for SOA Suite*.

Testing the Callout

After you have created the callout and upload its definition, you must test it to ensure it works as designed.

The high-level steps for testing a custom callout are:

1. Add the callout to a source or target and deploy the associated transfer. See [Adding the Callout to a Source](#) or [Adding the Callout to a Target](#).
2. After the transfer runs, look at the source or target report. See [Viewing the Report to Verify the Callout Action](#).
3. Update the callout if you need to correct errors or make improvements. See [Updating the Callout](#).

Note:

If you were logged into the MFT console when you ran the `createCallouts` command, you must log out and log in again before you can add the callout to a source or target.

Adding the Callout to a Source

The steps for adding the Newline Conversion callout to a source are:

1. Click the arrow to the left of **Sources** in the left pane navigator.
The sources are listed.
2. Click the source name or right-click it and then select the Open menu item.
The source tab opens.
3. Click the arrow to the left of **Actions**.
The Actions section opens.
4. Click **add processing actions**.
The Processing Actions dialog opens.
5. Select **New line conversion** from the All Actions drop-down list.
6. Click **Add to List**.
To remove an action from the list, click the Delete icon to the right of the action.
7. Type `Dos2Unix` or `Unix2Dos` in the **Type** text box.
8. Click **OK**.
To cancel adding actions, click **Cancel**.
9. **Save** and optionally **Deploy** the source.
10. Add the source to a transfer and **Deploy** the transfer.
See [Deploying and Testing Transfers](#) for more information.

Adding the Callout to a Target

The steps for adding the Newline Conversion callout to a target are:

1. Click the arrow to the left of **Transfers** in the left pane navigator.
The transfers are listed.
2. Click the transfer name or right-click it and then select the Open menu item.
The transfer tab opens.
3. Click the arrow to the left of the target.
The target settings are displayed.
4. Click **add preprocessing actions**.
The Pre-Processing Actions dialog opens.
5. Select **New line conversion** from the All Actions drop-down list.
6. Click **Add to List**.
To remove an action from the list, click the Delete icon to the right of the action.
7. Type `Dos2Unix` or `Unix2Dos` in the **Type** text box.
8. Click **OK**.
To cancel adding actions, click **Cancel**.
9. **Save** and **Deploy** the transfer.
See [Deploying and Testing Transfers](#) for more information.

Viewing the Report to Verify the Callout Action

To access the report for the source or target, go to the Monitoring page and see [Interpreting Source_ Transfer_ and Target Reports](#).

The **New line conversion** callout is listed in the Source Pre-Processing table in the source report or the Target Pre-Processing table in the target report. The Status column of either table shows the value `Processed` if the callout action was successful.

In addition, the Advanced section of the report includes `payloadModified=yes` in the Message Property field if the callout action was successful.

See [Source Reports](#) or [Target Reports](#) for more information.

Updating the Callouts

Keep these points in mind when updating callouts:

- No MFT server restart is needed when creating or updating callouts.
- If you change only the Java code, and the parameters are unchanged, all you need to do is copy the new JAR file into the callouts directory.
- If you change the definition file, and the parameters are unchanged, all you need to do is use the `updateCallouts` command to reload the callouts definition. `updateCallouts` schema is same as `createCallouts` schema without the `parameters` element.

Do not pass the parameter element when running the `updateCallouts` command.

updateCallouts Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://xmlns.oracle.com/mft"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mft="http://xmlns.oracle.com/mft"
  elementFormDefault="qualified">

  <xsd:element name="Callouts">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Callout" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:attribute name="name" use="required">
              <xsd:simpleType>
                <xsd:restriction base="mft:non-empty-string">
                  <xsd:maxLength value="128"/>
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:attribute>
            <!-- multiple values separated by comma are supported -->
            <xsd:attribute name="groupName" type="mft:non-empty-string"
              use="optional"/>
            <xsd:attribute name="implementationClass" type="mft:non-empty-
              string"/>
            <xsd:attribute name="libraryName" type="mft:non-empty-string"/>
            <xsd:attribute name="timeout" type="xsd:integer"
              use="optional" default="30"/>
            <xsd:attribute name="description" type="mft:non-empty-string"/>
            <xsd:attribute name="helpText" type="mft:non-empty-string"/>
          </xsd:complexType>
          <xsd:unique name="Callout_UK">
            <xsd:selector xpath="mft:Callout"/>
            <xsd:field xpath="@name"/>
          </xsd:unique>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:simpleType name="non-empty-string">
    <xsd:restriction base="string">
      <xsd:minLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

The following example shows how to increase the timeout:

```

<?xml version="1.0" encoding="UTF-8"?>
<mft:callouts xmlns:mft="http://xmlns.oracle.com/mft"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.oracle.com/mftcallout.xsd">
  <mft:Callout description="Custom New line conversion"
    helpText="Custom New line conversion"
    groupName="Source-pre,Target-pre" timeout="300"
    implementationClass="com.mft.sample.callout.NewlineConversion"
    libraryName="NewlineConversion.jar"
    name="NewlineConversion"/>
</mft:Callouts>

```

- If you add, delete, or modify callout parameters, you must perform these steps:

1. Delete the callout using the `deleteCallout` command.
2. Recreate the callout using the `createCallouts` command.
3. Reconfigure the source or target to use the new callout version.

For more information about MFT WLST commands, see [MFT WLST Command Summary](#) and MFT Callout Commands in *WLST Command Reference for SOA Suite*.

Reference Files

The reference files shows samples of some of the custom callouts.

This section lists the contents of the following files:

- [PreCalloutPlugin Interface](#)
- [PostCalloutPlugin Interface](#)
- [Callout Definition Schema](#)
- [PostDeliveryFailureCallout Login Interface](#)

PreCalloutPlugin Interface

Example - PreCalloutPlugin Interface shows the Java code for the `com.oracle.callout.PreCalloutPlugin` interface.

Example - PreCalloutPlugin Interface

```
public interface PreCalloutPlugin {

    /**
     * Depending on the return of this function output stream for changing the
     * pay load content will be passed. Only if the payload change is required
     * return true.
     *
     * @param context
     * @param calloutParams
     * @return
     */
    public boolean isPayloadChangeRequired(final PluginContext context,
        final Map<String, String> calloutParams);

    /**
     * This function will be called only if the isPayloadChangesRequired returns
     * true. This method must write final pay load content to the output stream.
     * Any custom properties or the header properties can be set at the context
     * level.
     *
     * @param context
     * @param input
     * @param out
     * @param calloutParams
     * @return
     */
    public PluginOutput process(final PluginContext context,
        final InputStream input, OutputStream out,
        final Map<String, String> calloutParams);

    /**
     * This function will be called only if the isPayloadChangesRequired returns
```

```

    * false. Any custom properties or the header properties can be set at the
    * context level.
    *
    * @param context
    * @param input
    * @param out
    * @param calloutParams
    * @return
    */
    public PluginOutput process(final PluginContext context,
                               final InputStream input, final Map<String, String> calloutParams);
}

```

PostCalloutPlugin Interface

Example - PostCalloutPlugin Interface shows the Java code for the `com.oracle.callout.PostCalloutPlugin` interface.

Example - PostCalloutPlugin Interface

```

public interface PostCalloutPlugin {

    /**
     * Any custom properties or the header properties can be set at the
     * context level.
     *
     * @param context
     * @param input
     * @param out
     * @param calloutParams
     * @return
     */
    public PluginOutput process(final PluginContext context,
                               final InputStream input, final Map<String, String> calloutParams);
}

```

Callout Definition Schema

The `callout.xsd` file defines the format of the callout definition XML file. **Callout Definition Schema** shows its contents.

Callout Definition Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://xmlns.oracle.com/mft"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mft="http://xmlns.oracle.com/mft"
  elementFormDefault="qualified">
  <xsd:element name="Callouts">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Callout" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element minOccurs="0" maxOccurs="unbounded"
                ref="mft:Parameter"/>
            </xsd:sequence>
            <xsd:attribute name="name" use="required">
              <xsd:simpleType>
                <xsd:restriction base="mft:non-empty-string">

```

```

        <xsd:maxLength value="128"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
<!-- multiple values separated by comma are supported -->
<xsd:attribute name="groupName" type="mft:non-empty-string"
use="optional"/>
    <xsd:attribute name="implementationClass" type="mft:non-empty-
string"/>
        <xsd:attribute name="libraryName" type="mft:non-empty-string"/>
    <xsd:attribute name="executeOnFailure" type="xsd:boolean" default="false"/>
    <xsd:attribute name="timeout" type="xsd:integer" use="optional"
default="30"/>
        <xsd:attribute name="description" type="mft:non-empty-string"/>
        <xsd:attribute name="helpText" type="mft:non-empty-string"/>
    </xsd:complexType>
    <xsd:unique name="Callout_UK">
        <xsd:selector xpath="mft:Callout"/>
        <xsd:field xpath="@name"/>
    </xsd:unique>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Parameter">
    <xsd:complexType>
        <xsd:attribute name="name" type="mft:non-empty-string" use="required"/>
        <xsd:attribute name="parameterType" type="mft:parameterTypeLOV"
            use="required"/>
        <xsd:attribute name="basic" type="boolean" use="optional" default="false"/>
        <xsd:attribute name="mandatory" type="boolean" use="optional"
            default="false"/>
        <xsd:attribute name="hidden" type="boolean" use="optional" default="false"/>
        <!-- multiple values separated by comma is supported -->
        <xsd:attribute name="groupName" type="mft:non-empty-string"
            use="optional"/>
        <!-- sub-groups can have multiple values separated by comma. Ex:
"source,target,listening,someOtherGroups" -->
        <xsd:attribute name="sub-groups" type="mft:non-empty-string"
            use="optional"/>
        <!-- VRule: defaultValue and parameterTypeLOV MUST match -->
        <xsd:attribute name="defaultValue" type="mft:non-empty-string"
            use="optional"/>
        <xsd:attribute name="listValue" type="mft:non-empty-string"
            use="optional"/>
        <xsd:attribute name="mapValue" type="mft:non-empty-string"
            use="optional"/>
        <!-- changed refValue from IDREF to non-empty-string as IDREF does not allow ",".
In the
        future refValue can have value separated by ", ". Ex: "xPathaParams,
someOtherParams" -->
        <xsd:attribute name="refValue" type="mft:non-empty-string" use="optional"/>
        <!--attribute name="tpOverrideable" type="boolean" use="optional" default="true"/--
    >
        <xsd:attribute name="overrideLevel" type="mft:overrideLevelLOV"
            use="optional" default="all"/>
        <xsd:attribute name="displayName" type="mft:non-empty-string"/>
        <xsd:attribute name="description" type="mft:non-empty-string"/>
        <xsd:attribute name="helpText" type="mft:non-empty-string"/>
    </xsd:complexType>
</xsd:element>
<xsd:simpleType name="overrideLevelLOV">

```

```

    <xsd:restriction base="string">
      <xsd:enumeration value="admin"/>
      <xsd:enumeration value="tp"/>
      <xsd:enumeration value="host"/>
      <xsd:enumeration value="tpa"/>
      <xsd:enumeration value="all"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="non-empty-string">
    <xsd:restriction base="string">
      <xsd:minLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="parameterTypeLOV">
    <xsd:restriction base="string">
      <xsd:enumeration value="float"/>
      <xsd:enumeration value="integer"/>
      <xsd:enumeration value="string"/>
      <xsd:enumeration value="boolean"/>
      <xsd:enumeration value="date"/>
      <xsd:enumeration value="SO"/>
      <xsd:enumeration value="MO"/>
      <xsd:enumeration value="credential"/>
      <xsd:enumeration value="list"/>
      <xsd:enumeration value="map"/>
      <xsd:enumeration value="ref"/>
      <xsd:enumeration value="hex"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

```

PostDeliveryFailureCalloutPlugin Interface

If there is a delivery failure, a target post callout is executed. **Example PostDeliveryFailureCalloutPlugin Interface** shows the Java code for the `postErrorCallout` plugin interface.

Example - PostDeliveryFailureCalloutPlugin Interface

```

package com.oracle.callout.sample;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.InputStream;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Map;

import oracle.tip.mft.bean.SourceMessage;
import oracle.tip.mft.bean.TargetMessage;
import oracle.tip.mft.engine.processor.plugin.PluginContext;
import oracle.tip.mft.engine.processor.plugin.PostDeliveryFailureCalloutPlugin;
import oracle.tip.mft.notification.message.EmailMessage;
import oracle.tip.mft.system.MFTConstant.ArtifactType;
import oracle.tip.mft.system.MFTUtil;

public class FailedTransferNotification implements
    PostDeliveryFailureCalloutPlugin {

```

```
private final String lineEnd = System.getProperty("line.separator");

@Override
public void process(PluginContext context, InputStream input,
    Map calloutParams) throws Exception {

    File emailTemplateFile = new File(
        calloutParams.get("Email Template"));
    String toAddress = calloutParams.get("Email Address");

    TargetMessage tgtMessage = (TargetMessage) context.getMessage();
    tgtMessage.getFileName();
    SourceMessage srcMessage = tgtMessage.getInstance()
        .getTransferInstance().getSourceMessage();

    if (!emailTemplateFile.exists()) {
        throw new RuntimeException("Email template file doesn't exists");
    }

    FileReader reader = new FileReader(emailTemplateFile);
    BufferedReader bufferIn = new BufferedReader(reader);

    String fromAddress = null;

    String emailSubject = null;
    StringBuffer emailBody = new StringBuffer();
    String line = null;
    try {
        while ((line = bufferIn.readLine()) != null) {
            line = line.trim();
            if (line.length() == 0) {
                // ignore empty line
            } else {
                if (line.toUpperCase().startsWith("FROM")
                    && line.indexOf("=") != -1) {
                    fromAddress = line.substring(line.indexOf("=") + 1)
                        .trim();
                    break;
                } else {
                    throw new RuntimeException("From address not found");
                }
            }
        }
    }

    while ((line = bufferIn.readLine()) != null) {
        line = line.trim();
        if (line.length() == 0) {
            // ignore empty line
        } else {
            if (line.toUpperCase().startsWith("SUBJECT")
                && line.indexOf("=") != -1) {
                emailSubject = line.substring(line.indexOf("=") + 1)
                    .trim();
                break;
            } else {
                throw new RuntimeException("Email subject not found");
            }
        }
    }
}
```

```
    }
  }
  boolean bodyElemFound = false;
  while ((line = bufferIn.readLine()) != null) {
    if (!bodyElemFound) {
      line = line.trim();
      if (line.length() == 0) {
        // ignore empty line
      }
      if (line.toUpperCase().startsWith("BODY")
          && line.indexOf("=") != -1) {
        emailBody = emailBody.append(line.substring(line
            .indexOf("=") + 1));
        bodyElemFound = true;
      } else {
        throw new RuntimeException("Email body not found");
      }
    } else {
      emailBody.append(line);
      emailBody.append(lineEnd);
    }
  }
}

} catch (Throwable e) {

  throw new RuntimeException(e);
} finally {
  bufferIn.close();
  reader.close();
}

String emailBodyStr = emailBody.toString();

String fileName = srcMessage.getFileName();
if( fileName == null){
  fileName = " ";
}
String userName = srcMessage.getSenderUserName();
if( userName == null){
  userName = " ";
}

String sourceName = srcMessage.getSourceName();
double fileSizeInMB = srcMessage.getDataStorage().getPayloadSize() / 1048576.0;
Calendar cal = Calendar.getInstance();
DecimalFormat df = new DecimalFormat("#.####");
String time = new SimpleDateFormat("dd MMM yyyy HH:mm:ss").format(cal.getTime());
String transferUrl = MFTUtil.getInstance().getReportTrackingURL(
    ArtifactType.TRANSFER,
    tgtMessage.getInstance().getTransferInstance().getId(), null);
// replace all place holders
emailSubject = emailSubject.replaceAll("%FILENAME%", fileName);
String errorText = "";
if( tgtMessage.getErrorInfo().getErrorTextClob() != null ){
  errorText = tgtMessage.getErrorInfo().getErrorTextClob().toString();
}

String errorDesc = "";
if( tgtMessage.getErrorInfo().getErrorDescriptionClob() != null ){
```

```

        errorDesc = tgtMessage.getErrorInfo().getErrorDescriptionClob().toString();
    }

    emailBodyStr = emailBodyStr.replaceAll("%FILENAME%", fileName)
        .replaceAll("%FILESIZE%", df.format(fileSizeInMB))
        .replaceAll("%USER%", userName).replaceAll("%DATE%", time)
        .replaceAll("%SOURCE%", sourceName)
        .replaceAll("%ERRORDESCRIPTION%", errorDesc)
        .replaceAll("%ERRORCODE%", tgtMessage.getErrorInfo().getErrorCode())
        .replaceAll("%ERRORTEXT%", errorText )
        .replaceAll("%TRANSFERURL%", transferUrl);

    EmailMessage emailMessage = new EmailMessage();
    emailMessage.setFromEmailAddress(fromAddress);
    emailMessage.setReplyToEmailAddress(fromAddress);
    emailMessage.setReceiverEmailAddress(toAddress);
    emailMessage.setSubject(emailSubject);
    emailMessage.setMessageContent(emailBodyStr);
    emailMessage.setMimeType("text/html");
    context.sendEmailNotification(emailMessage) ;

}

public String convertWildcardToRegex(String express1) {
    String outstr = "";
    int strlen = express1.length();
    for (int ii = 0; ii < strlen; ii++) {
        char testch = express1.charAt(ii);
        if (testch == '.') {
            outstr = outstr + "\\.";
        } else if (testch == '*') {
            outstr = outstr + ".*";
        } else if (testch == '?') {
            outstr = outstr + ".";
        } else {
            outstr = outstr + testch;
        }
    }
    return outstr;
}
}
}

```

Validating using Custom Callout

You can validate a custom callout action. For example, when validating checksum, you can validate by setting the exception object in the plugin output in the callout implementation.

The code for validation is as follows:

```

public PluginOutput process(PluginContext context, InputStream input,
Map<String, String> calloutParams) {
    PluginOutput res = new PluginOutput() ;
    boolean isValid = validateChecksum(context, input, calloutParams) ;
    if( ! isValid ){res.setException(new Exception("Checksum mismatch, corrupted
payload.")) ;
}
}

```

```
return res ;  
}
```

The checksum action status for the above example will be failed for the instance report page.
The message status will be Failed[Pre-processing].

4

Integrating Oracle Managed File Transfer with Other Products

Learn how to use Oracle Managed File Transfer (MFT) in various application integrations.

This chapter includes the following sections:

- [Compatible Technologies and Integration Strategies](#)
- [Managing Domains](#)
- [Integrating with Oracle SOA Suite](#)
- [Integrating with Oracle Service Bus](#)
- [Integrating with Oracle B2B](#)
- [Integrating with Oracle Data Integrator](#)
- [Integrating with Web Services](#)
- [Integrating with Oracle WebCenter Content](#)
- [Integrating with Oracle Storage Cloud Service](#)
- [MFT WSDL Files](#)

Compatible Technologies and Integration Strategies

Oracle Managed File Transfer is compatible with various technologies and can be integrated using different integration strategies.

Table 4-1 Compatible Technologies and Integration Strategies for Oracle Managed File Transfer

Technology	Source and Target Types	Integration Strategy
File system	File, FTP Remote, sFTP Remote, FTP Embedded, sFTP Embedded	Use standard file management and FTP or sFTP commands.
Web services	SOAP, SOA, Service Bus, ODI	Configure the integrating application to recognize the source or target as a web service.
B2B	B2B	Configure the integrating application to recognize the source or target as a trading partner channel or endpoint.

Within each technology, each integrating application recognizes Oracle Managed File Transfer differently. Here are some examples:

- A SOA source in Oracle Managed File Transfer corresponds to an MFT reference binding component in SOA.

- A SOA target in Oracle Managed File Transfer corresponds to an MFT service binding component in SOA.
- A Service Bus source in Oracle Managed File Transfer corresponds to a business service in Oracle Service Bus.
- A Service Bus target in Oracle Managed File Transfer corresponds to a proxy service in Oracle Service Bus.

Some integrating applications, such as SOA, recognize Oracle Managed File Transfer with a specific component or setting. Others, such as Oracle Service Bus, recognize Oracle Managed File Transfer generically according to the technology.

Managing Domains

You must configure domains for SOA, Service Bus, B2B, and ODI sources and targets if these integrating applications are not collocated on the same Oracle WebLogic Server instance or cluster with Oracle Managed File Transfer.

The steps to configure domains are:

1. Open the Domains tab on the Administration page.

2. Click the **Add** icon.

An empty row is added to the domain table.

3. Enter the following information in the table cells:

- Domain Alias: The host name for connecting to the domain.

The **Domain Alias** setting for a source or target maps to this alias.

- Connection URL: The service endpoint URL for connecting to B2B applications running on remote servers. These servers can be in the same Oracle WebLogic Server domain as Oracle Managed File Transfer or in a different domain. It is used to send messages to B2B targets.
- User: The user as whom to access the domain.
- Password: The user password.
- Confirm Password: The user password confirmation.
- Tracking URL: The URL for the source or target location from or to which files are transferred, if different from the **Connection URL**. See [Specifying the Tracking URL](#) for more information.
- Type: The domain type, B2B, SOA, Service Bus, or ODI.
This is only an optional label. It does not determine which domains are available for which source and target types.
- Description: A text description of the domain.
- Example: t3://host:port (host and port details of the WLS Server where B2B is running).

4. Click **Show Domain Details** to display a list of referring sources and targets for the selected domain.

By default, all source and target types are listed. You can uncheck **B2B**, **SOA**, **Service Bus**, or **ODI** to hide sources and targets of that type.

5. Click **Save**.

To delete a domain, select the table row and click the **Delete** icon. To cancel all changes since you last saved, click **Revert**.

After you create sources and targets that reference the domain aliases, you can select a domain and click **Show Domain Details** to see the referring sources and targets.

Specifying the Tracking URL

The **Tracking URL** is the URL for the source or target location from or to which files are transferred, if different from the **Connection URL**.

For all domain types, the tracking URL is the console URL used to access the respective reporting module for the product being integrated. This URL can be the load balancer IP or server address. It is used in generating the dynamic URL to open the respective reporting page from the Oracle Managed File Transfer monitoring module. For examples, see [Interlinked SOA and MFT Reports](#), [Interlinked B2B and MFT Reports](#)

When you select a **Domain Alias** for a source or target, all configured domains are available for B2B targets, but only domains with a tracking URL are available for B2B sources and for SOA, Service Bus, and ODI sources and targets.

The connection URL for B2B remote integration points to the B2B managed server. However, remote B2B report integration may use the HTTP URL of a load balancer instead, in which case you must specify a tracking URL.

The connection URL for Oracle SOA Suite points to the SOA managed server. However, because Oracle Enterprise Manager Fusion Middleware Control runs on the administration server, the URL for flow trace navigation is different. Therefore, to ensure that report interlinking works as described in [Interlinked SOA and MFT Reports](#), you must specify a tracking URL.

Tracking URL syntax is as follows:

```
http://host:port#domain_name#domain_type
```

The default for the optional *domain_type* is `weblogic_domain`.

For example, if the MFT console URL is `mft.example.com:8001` and both MFT and Fusion Middleware Control for SOA are in the same `mft_domain`, use this as the tracking URL:

```
http://mft.example.com:8001#mft_domain
```

If the Fusion Middleware Control console URL is `soa.example.com:21374` and Fusion Middleware Control is in the `WLS_SOA` domain, use this as the tracking URL:

```
http://soa.example.com:21374#WLS_SOA
```

Integrating with Oracle SOA Suite

Oracle JDeveloper provides a specialized MFT component under BPEL Services to represent an Oracle Managed File Transfer source or target in a BPEL process. After creating a source or target of type SOA in Oracle Managed File Transfer, create a corresponding MFT component in your SOA application.

Integration between SOA and MFT uses several WSDL files. See [MFT WSDL Files](#) for more information.

You must configure a SOA domain in Oracle Managed File Transfer if Oracle SOA Suite is not collocated with MFT. To ensure that report interlinking works as described in [Interlinked SOA](#)

and [MFT Reports](#), you must specify a tracking URL. For more information, see [Specifying the Tracking URL](#).

Creating an MFT Reference for a SOA Source

A SOA source in Oracle Managed File Transfer corresponds to an MFT reference binding component in SOA. The SOA application sends the file.

The steps for this process are:

1. Create a new SOA Application and project in Oracle JDeveloper.
2. With the project open, display the Technology section of the Component Palette.
3. Drag and drop the MFT component icon into the right Partner Link swimlane. The MFT Configuration Wizard opens.
4. On the MFT Adapter Reference page, specify a name or accept the default name of `mftReferencenumber`. **Reference** is selected as the binding. Click **Next**.
5. On the Adapter Interface page, select **Define using a new MFT Reference**. Click **Next**.
6. On the Service Connection page, select the connection to the Oracle WebLogic Server on which Oracle Managed File Transfer is installed from the **AppServer Connection** drop-down list. Select the managed server on which Oracle Managed File Transfer runs from the **SOA Server** drop-down list if it does not autofill. Click **Test MFT** to test the communication with Oracle Managed File Transfer. Click **Next**.
7. On the Reference Configuration page, select the MFT source name from the **Source** drop-down list. The endpoint location autofills in the **Endpoint** field. Click **Finish**.
8. The MFT component appears in the BPEL process. Work with it as you would any external reference.

Creating an MFT Service for a SOA Target

A SOA target in Oracle Managed File Transfer corresponds to an MFT service binding component in SOA. The SOA application receives the file.

The steps for this process are:

1. Create a new SOA Application and project in Oracle JDeveloper.
2. With the project open, display the Technology section of the Component Palette.
3. Drag and drop the MFT component icon into the left Partner Link swimlane. The MFT Configuration Wizard opens.
4. On the MFT Adapter Reference page, specify a name or accept the default name of `mftServicenumber`. **Service** is selected as the binding. Click **Finish**.
5. The MFT component appears in the BPEL process. Work with it as you would any exposed service.

Interlinked SOA and MFT Reports

An Oracle Managed File Transfer instance report for a SOA source or target has a Correlation Flow ID link. Clicking this link opens the corresponding Flow Trace page in Oracle Enterprise Manager Fusion Middleware Control for the SOA instance.

Likewise, a Fusion Middleware Control Flow Trace for Oracle Managed File Transfer has a Managed File Transfer link in the Trace section. Clicking this link opens the corresponding Oracle Managed File Transfer instance report.

To ensure that report interlinking works when Oracle SOA Suite is not collocated with MFT, you must specify a tracking URL. For more information, see [Specifying the Tracking URL](#).

See [Interpreting Source_ Transfer_ and Target Reports](#) for more information about Oracle Managed File Transfer instance reports.

Integrating with Oracle Service Bus

Oracle Service Bus uses a business service for outgoing data and a proxy service for incoming data. The corresponding endpoints in Oracle Managed File Transfer are a Service Bus source and a Service Bus target, respectively.

You must configure a Service Bus domain in Oracle Managed File Transfer if Oracle Service Bus is not collocated with MFT. For more information, see [Managing Domains](#).

Creating a Business Service for a Service Bus Source

A Service Bus source in Oracle Managed File Transfer corresponds to a business service in Oracle Service Bus. Oracle Service Bus calls Oracle Managed File Transfer and Oracle Service Bus sends the file. You can create a business service from either Oracle JDeveloper or the Oracle Service Bus console.

For more information, see *Creating and Configuring Business Services in Developing Services with Oracle Service Bus*.

Creating a Business Service in Oracle JDeveloper

The steps for this process are:

1. Open or create the application and project to which you want to add the business service.
2. In the Application Navigator, right-click the project and select **New > Business Service** to display the Create Business Service wizard. On wizard pages not mentioned, you can accept the default values and click **Next**.
3. On the Business Service Create Service Page, type a name, and set the **Definition** to WSDL.
4. Click **Browse**. Navigate to the `WLS_Home/mft/integration/wsd1` directory and select the `MFTSOAService.wsdl` file. Click **OK**.
See [MFT WSDL Files](#) for more information about this file.
5. The WSDL settings appear on the Create Service Page. Click **Next**.
6. On the Business Service Transport Configuration Page, set the **Protocol** to `http`. Set the **Endpoint URI** to match the **URL** setting in the Service Bus source. Click **Create**.
7. The Business Service Definition Editor appears with the general configuration of the new business service displayed.

Creating a Business Service in the Oracle Service Bus Console

The steps for this process are:

1. Import the WSDL file for Oracle Managed File Transfer into your Oracle Service Bus project. The full path to this file is `WLS_Home/mft/integration/wsd/ MFTSOAService.wsdl`. See [MFT WSDL Files](#) for more information about this file.
For instructions on the Service Bus side, see Importing and Exporting Resources and Configurations in *Developing Services with Oracle Service Bus*.
2. In the Resource panel, select the project to which you want to add a business service, and then click the down arrow next to the **Create** icon.
3. Select **Business Service** to display the Create Business Service wizard.
4. In the Create Service section of the Create page, enter a name for the business service.
5. In the Service Definition section of the Create page, select `WSDL Based Service`.
6. Click the **Search** icon to search for a WSDL resource. With all fields blank, click **Search**. This finds all imported WSDL files. Select the WSDL file for Oracle Managed File Transfer in the search results table. Click **OK**.
7. Once you specify the WSDL file, select the port or binding to use from the **Port/Binding** field. Click **Next**.
8. On the Transport Page, set the **Protocol** to `http`. Set the **Endpoint URI** to match the **URL** setting in the Service Bus target. Click **Create**.
9. The Business Service Definition Editor appears with the general configuration of the new business service displayed.

Creating a Proxy Service for a Service Bus Target

A Service Bus target in Oracle Managed File Transfer corresponds to a proxy service in Oracle Service Bus. Oracle Managed File Transfer calls Oracle Service Bus and Oracle Service Bus receives the file. You can create a proxy service from either Oracle JDeveloper or the Oracle Service Bus console.

For more information, see *Creating and Configuring Proxy Services in Developing Services with Oracle Service Bus*.

Creating a Proxy Service in Oracle JDeveloper

The steps for this process are:

1. Open or create the application and project to which you want to add the proxy service.
2. In the Application Navigator, right-click the project and select **New > Proxy Service** to display the Create Proxy Service wizard. On wizard pages not mentioned, you can accept the default values and click **Next**.
3. On the Proxy Service Create Service Page, type a name, and set the **Definition** to `WSDL`.
4. Click **Browse**. Navigate to the `WLS_Home/mft/integration/wsd` directory and select the `MFTAnyTypeService.wsdl` file. Click **OK**.
See [MFT WSDL Files](#) for more information about this file.
5. The WSDL settings appear on the Create Service Page. Click **Next**.
6. On the Proxy Service Transport Configuration Page, set the **Protocol** to `ws`. Set the **Endpoint URI** to match the **URL** setting in the Service Bus target. Click **Create**.
7. The Proxy Service Definition Editor appears with the general configuration of the new proxy service displayed.

Creating a Proxy Service in the Oracle Service Bus Console

The steps for this process are:

1. Import the WSDL file for Oracle Managed File Transfer into your Oracle Service Bus project. The full path to this file is `WLS_Home/mft/integration/wsdl/MFTAnyTypeService.wsdl`.
See [MFT WSDL Files](#) for more information about this file.
For instructions on the Service Bus side, see [Importing and Exporting Resources and Configurations in *Developing Services with Oracle Service Bus*](#).
2. In the Resource panel, select the project to which you want to add a proxy service, and then click the down arrow next to the **Create** icon.
3. Select **Proxy Service** to display the Create Proxy Service wizard.
4. In the Create Service section of the Create page, enter a name for the proxy service.
5. In the Service Definition section of the Create page, select `WSDL Based Service`.
6. Click the **Search** icon to search for a WSDL resource. With all fields blank, click **Search**. This finds all imported WSDL files. Select the WSDL file for Oracle Managed File Transfer in the search results table. Click **OK**.
7. Once you specify the WSDL file, select the port or binding to use from the **Port/Binding** field. Click **Next**.
8. On the Transport Page, set the **Protocol** to `ws`. Set the **Endpoint URI** to match the **URL** setting in the Service Bus target. Click **Create**.
9. The Proxy Service Definition Editor appears with the general configuration of the new proxy service displayed.

Integrating with B2B

B2B recognizes Oracle Managed File Transfer as a Remote Trading Partner. B2B uses an external delivery channel to send files to Oracle Managed File Transfer.

B2B requires no special configuration to receive files. Instead, you configure a B2B domain in Oracle Managed File Transfer.

Creating a Remote Trading Partner Channel for a B2B Source

A B2B source in Oracle Managed File Transfer corresponds to an external delivery channel for a Remote Trading Partner in B2B. You can either configure a B2B domain in Oracle Managed File Transfer (see [Managing Domains](#)) or configure a channel in B2B.

The steps for this process are:

1. Go to the Partners page in B2B.
2. Select the remote trading partner in the Partner pane, or create a new remote trading partner.
3. Select the Channels tab.
4. Type a **Name** for the new channel.
5. Select `Generic MFT-1.0` from the **Protocol** drop-down list.

6. On the Transport Control Parameters tab, set the following parameters:
 - URL: The URL for connecting to Oracle Managed File Transfer.
 - User name: The user name for connecting to Oracle Managed File Transfer.
 - Password, Confirm Password: The password for connecting to Oracle Managed File Transfer.
 - Source: The name of the B2B source in Oracle Managed File Transfer.
 - Transfer: The name of the transfer associated with the B2B source. This is optional.
 - Target: The name of the target associated with the B2B source. This is optional.

Source is always required. **URL**, **User name**, and **Password** are required if B2B must connect to Oracle Managed File Transfer on a different server.

7. Click **Save**.
8. The external delivery channel is ready to send files to Oracle Managed File Transfer.

For more information, see *Configuring Channels* in *User's Guide for Oracle B2B*.

Configuring a B2B Domain for a B2B Target

A B2B target in Oracle Managed File Transfer calls the EJB for the Host Trading Partner in B2B. No B2B configuration is needed. However, you must configure a B2B domain in Oracle Managed File Transfer if B2B is not collocated. For more information, see [Managing Domains](#).

Interlinked B2B and MFT Reports

An Oracle Managed File Transfer instance report for a B2B source or target has a Correlation Flow ID link. Clicking this link opens the corresponding Business Message report for the B2B instance.

Likewise, a B2B Business Message report for Oracle Managed File Transfer has an MFT Flow link in the Flow Trace column of the Result table. Clicking this link opens the corresponding Oracle Managed File Transfer instance report.

For more information about Oracle Managed File Transfer instance reports, see [Interpreting Source, Transfer, and Target Reports](#).

Integrating with Oracle Data Integrator

You can integrate Oracle Managed File Transfer with Oracle Data Integrator (ODI) using different integration methods.

Oracle Managed File Transfer can integrate with Oracle Data Integrator (ODI) in one of two ways:

- Through the file FTP or sFTP.
Using the file system is the simplest and most common integration method.
- Through web services, using the `OdiInvokeWebService` tool for ODI sources or configuring data services for ODI targets.

For the web service integration method, you must configure an ODI domain in Oracle Managed File Transfer if Oracle Data Integrator is not collocated with MFT. For more information, see [Managing Domains](#).

Using the file system is best for large files and batch processing, while using web services is more suited to smaller payloads for updates or trickle feeds.

Using the File System and FTP and SFTP for ODI Integration

When you use the file FTP and SFTP for ODI integration, you use ODI source or target types by selecting the binding type as File, FTP Remote and SFTP Remote. The use cases are as follows:

ODI to MFT: A trigger event service is exposed as part of the trigger-based content delivery feature. ODI invokes that service by providing the ODI/JCA source name and other required parameters and in response a unique MFT id "EventSessionId" is returned. The source is validated and verified for Event invocation and then file transfer is initiated. ODI can query the status of MFT instances by invoking another operation "getInstanceDetails" in the same trigger service providing the EventSessionId returned earlier.

MFT to ODI: The `odiInvoke` service acts in a post processing function. This post processing function is used to configure the `odiInvoke` service and is invoked after delivering the payload to the JCA bindings configured at the ODI target. The `odiInvoke` service retrieves the payload from the JCA binding target type configured at ODI target which and this retrieving is outside the scope of MFT. The MFT message is marked as complete when payload delivery is complete and the `odiInvoke` service is invoked.

These ODI transfers can use MFT features such as:

- Preprocessing, including encryption, decryption, and custom callouts that modify the payload
- Scheduling
- Resubmission

Invoking a Web Service for an ODI Source

An ODI source in Oracle Managed File Transfer is exposed as a web service that ODI can invoke using the `OdiInvokeWebService` tool.

Using an ODI source has the following limitations:

- MFT doesn't support synchronous invocation.
- The response MFT sends to ODI includes only the MFT message ID.
- MFT doesn't notify ODI of whether the transfer was successful.

However, you can create a custom callout to send a notification. See [Processing Transfers with Custom Callouts](#) for more information.

The steps for this process are:

1. In MFT, create an ODI source and enter a URL for the location.
2. In ODI, in the **Projects** tree in the Designer Navigator, select the project and package that will send a file to MFT.
3. Create an `OdiInvokeWebService` tool step in the package and select it.
4. Click the General tab under Properties.
5. Specify values for the following parameters:
 - **WSDL URL** — Specify the URL of the MFT ODI source WSDL file. This is the ODI source URL with a `?WSDL` suffix. For example:

```
http://www.example.com:7001/mftapp/services/transfer/odi-src-name?WSDL
```

See [MFT WSDL Files](#) for more information.

- **Port Type** — Specify `MFTServicePortType`.
- **Operation** — Specify `submit`.
- **XML Request** — Enter the web service payload, which includes SOAP headers and the file to be transferred.

6. Select File > **Save** to save the package.

See Using the `OdiInvokeWebService` Tool in *Developing Integration Projects with Oracle Data Integrator* for more information.

Creating a Data Service for an ODI Target

An ODI target in Oracle Managed File Transfer corresponds to a data service in ODI. ODI data services are specialized web services that provide access to data in datastores. ODI generates these data services and deploys them to the web services container in Oracle WebLogic Server.

Using an ODI target has the following limitations:

- Input from the MFT source must adhere to the ODI data service schema.
- The message type must be XML and must match the ODI WSDL format. MFT cannot perform this validation.
- MFT doesn't support synchronous messaging.
- MFT cannot access data exposed by the ODI data services.
- The Attachment **Payload Type** and the By Reference **Delivery Method** are not supported.
- Small payloads, not large files, are recommended.

The steps for this process are:

1. In ODI, in the **Models** tree in the Designer Navigator, select the model that contains the datastore.
2. Click the model to edit it.
3. Select **Services**.
4. Specify values for the following properties:
 - **Application server** — Specify the Oracle WebLogic Server host and port. For example:

```
localhost:7003
```
 - **Namespace** — Specify the MFT namespace:

```
http://xmlns.oracle.com/fmw/mft/soap
```
 - **Package name** — Specify the package name:

```
com.oracle.mft
```
 - **Name of Data Source** — Enter a data source name.
 - **Name of Data Service** — Enter a data service name.
 - **Active** — Check the Active box next to the datastore to be exposed as a web service.

Note the **Web Service name** and the **Published entity**. You need these to configure the MFT target.

5. Select File > **Save** to save the model.

See *Generating and Deploying Data Services in Administering Oracle Data Integrator* for more information.

After the data service is deployed in Oracle WebLogic Server, an MFT ODI target can invoke the change operations exposed by the data service.

The steps for this process are:

1. In MFT, create a target of type ODI.
2. Specify the data service URL as the MFT target **URL**. For example:

```
http://www.example.com:15101/model-name/web-svc-name
```
3. Click the arrow to the left of **Advanced Properties** to display the properties.
4. Specify values for the following properties:
 - **Service** — Specify the **Web Service name** from the ODI model.
 - **Port** — Specify the port. By default, this is the **Web Service name** with a `SoapPort` suffix.
 - **Action** — Specify the web service method used to retrieve the transferred file. By default, this is the **Published entity** from the ODI model with an `add` prefix and a `List` suffix.

You can display web service information in Oracle WebLogic Server or Oracle Enterprise Manager Fusion Middleware Control. See *Developing JAX-WS Web Services in Developing JAX-WS Web Services for Oracle WebLogic Server* for more information.

- **Payload Type** — Select **Inline**.
 - **Message Type** — Select **XML**.
5. **Save** the target.
 6. Add the target to a transfer.
 7. In the transfer configuration, click the arrow to the left of **Delivery Preferences** to display the options.
 8. Specify **Inline** as the **Delivery Method**.
 9. **Save** and **Deploy** the transfer.

Integrating with Web Services

Any application that can integrate with web services can integrate with Oracle Managed File Transfer. Sources and targets of type SOAP are exposed as standard web service endpoints.

The source and target settings specified in Oracle Managed File Transfer are the same settings you specify in the integrating application to allow it to recognize the source or target. The file `MFTAnyTypeService.wsdl` is for any type of application that integrates with web services. See [MFT WSDL Files](#) for more information.

For more information about web services, see *Overview of Web Services in Oracle Fusion Middleware 12c in Understanding Web Services*.

Integrating with Oracle WebCenter Content

By using Oracle Managed File Transfer, you can download and upload data from an Oracle content server using the Remote Intradoc Client (RIDC) client jar. The jar is provided by the Oracle WebCenter Content to connect and perform various actions.

RIDC provides a thin communication API for communication with Oracle WebCenter Content Server. This API removes data abstractions to the Oracle WebCenter Content Server while still providing a wrapper to handle connection pooling, security, and protocol specifics. RIDC supports Intradoc socket-based communication and the HTTP protocol.

Using the HTTP protocol to connect to WebCenter Content via the RIDC client, MFT supports the following properties for the HTTP protocol:

- ConnectionUrl
- Username
- Password
- Proxy server name
- Proxy user name
- isSecure
- socketTimeout
- connectionWaitTime
- contentType
- useSystemProxy

MFT exposes three content server services:

- GET_FILE: Service that returns a specific rendition of a content item revision. A copy of the file is retrieved without performing a checkout.
- CHECKIN_UNIVERSAL: Service that performs an Oracle Content Server-controlled check in.
- GET_SEARCH_RESULTS: Service that returns a list of content items that match specific search criteria.

For complete information on Oracle WebCenter Content, see *Developing with Oracle WebCenter Content*.

Integrating with Oracle Storage Cloud Service

Oracle Storage Cloud Service is part of the Infrastructure as a Service (IaaS) offerings. It is an enterprise-grade, large-scale object storage solution for files and unstructured data. Stored customer data is automatically stored to prevent data loss. You can use the service to back up content to an offsite location, programmatically store content, and share content with peers.

In Oracle Storage Cloud Service, data is stored in the form of objects. An object is most commonly created by uploading a file, although it can also be created from ephemeral unstructured data. Objects must be created within a container. A container is a user-created resource, which can hold an unlimited number of objects. Containers however, cannot store other containers. Both objects and containers can have custom metadata associated with them.

For more information about Oracle Storage Cloud Service, see [About Oracle Storage Cloud Service](#) in *Oracle Storage Cloud Service*.

Using MFT you can upload and download data between cloud and on-premise locations. Oracle MFT can be used to automate those transfers.

MFT WSDL Files

Oracle Managed File Transfer provides WSDL files, which integrating applications can use to create web service interfaces to Oracle Managed File Transfer.

These files are located in `WLS_Home/mft/integration/wSDL`. [Table 4-2](#) describes these files.

Table 4-2 MFT WSDL Files

WSDL File Name	Purpose	Operations	Usage
MFTAnyTypeService.wsdl	Used in MFT SOAP sources and targets. Accepts SOAP messages, including generic messages.	submitReference submitInline Both operations respond with the MFT Message ID and other tracking parameters.	For submitReference, the FTP or FILE reference is required in the MFTServiceInput element of the SOAP body. The TargetFileName and contentIdentifier are optional. For submitInline, the payload can be any valid XML code embedded directly under the SOAP body.
MFTService.wsdl	Used in MFT SOA sources and targets. Accepts SOA messages.	submit Responds with the Message ID and other tracking parameters.	An FTP or FILE reference or an inline or binary payload is required in the MFTServiceInput element of the SOAP body. The TargetFileName and contentIdentifier are optional.
MFTSOAService.wsdl	An abstract WSDL for Oracle JDeveloper integration. Used for creating the MFT reference WSDL.	submit Responds with the Message ID and other tracking parameters.	Same as MFTService.wsdl.
MFTDiscoveryService.wsdl	Used by Oracle JDeveloper to query all MFT SOA sources. Used during MFT SOA source selection, MFT SOA reference design, and SOA composite application building.	getSources Responds with the list of sources matching the type.	A single string parameter representing the source type is required in the MFTServiceInput element of the SOAP body. Allowed values for MFTServiceInput are SOAP, SOA, OSB (for Service Bus), or ODI. ODI can have a subtype of FILE. FTP, or SFTP. ODI can invoke this operation to list all ODI sources with JCA sources in the definition.

Table 4-2 (Cont.) MFT WSDL Files

WSDL File Name	Purpose	Operations	Usage
MFTResubmitService.wsdl	Used to support resubmission of MFT source, transfer, and target messages.	resubmitMessage Responds with true or false indicating success or failure.	Required input parameters are MessageType and MessageID. Allowed values for MessageType are SOURCE, TARGET, or TRANSFER. User comments are optional.

Headers of these WSDL files appear in the Source Report and Target Report for a transfer, which are accessible from the Monitoring page. For more information about these reports, see:

- [Interpreting Source, Transfer, and Target Reports](#)
- Source Report and Target Report in the *MFT Composer Online Help*.

The MFT WSIL URL is as follows:

```
http://host:port/inspection.wsil/?appname=mft-app
```

The MFT Discovery Service URL is as follows:

```
http://host:port/mftapp/services/MFTDiscoveryService?WSDL
```

 **Note:**

Some SOAP clients automatically generate an `input` tag in the WSDL file, which MFT interprets as a payload and appends to the transferred file. If this tag is not needed, you can delete it.

 **Note:**

Make sure the WSDL file has a valid value for `MFTServicePort` or you might see this error message:

```
oracle.jdeveloper.webservices.model.WebServiceException: Service
MFTService_soap contains no SOAP ports and cannot be used.
```

5

Monitoring Oracle Managed File Transfer

Learn to use the Monitoring page of the Oracle Managed File Transfer console. The Monitoring page displays information about deployments of source, target, and transfer artifacts, the page also displays dashboards for all artifacts, for all artifacts of a specific type (source, transfer, or target), or for an individual artifact.

Dashboards provide information about in-progress and completed file delivery instances. This chapter includes the following sections:

- [Monitoring Deployed Sources, Targets, and Transfers](#)
- [Monitoring Transfer Flows Using the Main Dashboard](#)
- [Interpreting Dashboards for All Transfers, Sources, or Targets](#)
- [Interpreting Single Artifact Transfer, Source, and Target Dashboards](#)
- [Interpreting Source, Transfer, and Target Reports](#)
- [Pause and Restart a Transfer](#)
- [Resubmit a Transfer](#)
- [Diagnose File Delivery Failures](#)
- [Diagnose Transfer Errors](#)

Monitoring Deployed Sources, Targets, and Transfers

After you deploy source, transfer, or target artifacts, you can view them in the Deployments tab on the Monitoring page.

You can select which artifacts to view:

- **Order by** time deployed, time updated, name, or artifact type, ascending or descending.
- Select a range using **Deployed** or **Updated** and **From** or **To** with dates and times.
- **Select, Unselect, or Invert All.** Inverting means selecting unselected artifacts and deselecting selected ones.
- **Show Only Sources, Targets, or Transfers.** You can show any or all types.
- **Show Deployment Details** shows labels and dependent artifacts for the selected artifact. The label is the name of the deployment corresponding to the artifact on the Oracle WebLogic Server managed server for Oracle Managed File Transfer. Artifacts associated with the same transfer are mutually dependent.

Note:

The Created timestamp refers to the creation of the label, not the artifact. Editing and redeploying an artifact creates a new label.

To refresh the display according to these selections, click the **Filter** or **Go** icon. The **Go** icon appears when you change these selections. To clear these selections and display according to the defaults, click the **Erase** icon.

Disabling Sources, Targets, and Transfers

All deployed artifacts are enabled by default. Disabling an artifact makes it inactive without undeploying it.

The steps for this process are:

1. Check the box to the left of the artifact.
2. Click the **Disable** button.

The Disable Deployments dialog appears, showing dependent artifacts affected by the proposed disabling. You can enter an optional comment.

3. Click the **Yes** button.

The Disable Deployments dialog shows whether disabling was successful.

4. Click the **OK** button.

If disabling was successful, the Status column entry for the artifact changes from Enabled in green to Disabled in red. Dependent artifacts may also be shown as disabled.

Reenabling Sources, Targets, and Transfers

You can reenable a disabled artifact.

The steps for this process are:

1. Check the box to the left of the artifact.
2. Click the **Enable** button.

The Enable Deployments dialog appears. It displays a confirmation dialog about associated artifacts. Select the **Enable all associated Transfers** checkbox to enable all the associated transfers with the selected source. Or leave the checkbox blank to enable only the selected source. You can enter an optional comment in **Comment for Enabling Deployments (Optional)** textbox.

3. Click the **Yes** button.

The Enable Deployments dialog shows whether enabling was successful.

4. Click the **OK** button.

If enabling was successful, the Status column entry for the artifact changes from Disabled in red to Enabled in green.

Note:

If dependent artifacts were disabled, you must reenable them separately. For example, if you disabled a source and the associated transfer was disabled, reenabling the source does not reenable the transfer.

Undeploying Sources, Targets, and Transfers

If you want to both inactivate and edit an artifact, undeploying it might be preferable to disabling it. Oracle Managed File Transfer maintains versions of its artifacts (source, transfer and target). When an artifact is deployed, the current version of the artifact is deployed. These versions are stored for future auditing. When you undeploy an artifact, the following occurs:

- The artifact becomes eligible for purging using a WLST command as described in [Oracle Managed File Transfer Utilities](#).
- That version cannot be deployed again. You cannot roll it back to previous version.

The steps for this process are:

1. Check the box to the left of the artifact.
2. Click the **Undeploy** button.

The Remove Deployments dialog appears. You can enter an optional comment.

3. Click the **Yes** button.

The Remove Deployments dialog shows whether undeployment was successful.

4. Click the **OK** button.

If undeployment was successful, the artifact is removed from the Deployments tab.



Note:

You cannot undeploy a source or target if the associated transfer is deployed. The Undeploy button remains grayed out.

Redeploying Sources, Targets, and Transfers

To redeploy an artifact, open its tab on the Design page and click **Deploy**. See [Deploying a Source, Target, or Transfer](#).

Versioning Sources, Targets, and Transfers

Oracle MFT uses MDS to store metadata. The MDS Repository maintains versions of the documents in a database-based repository. Versioning allows changes to metadata objects to be stored as separate versions rather than simply overwriting the existing data in the metadata repository. It provides version history, as well as the ability to label versions so that you can access the set of metadata as it was at a given point in time. An artifact can be modified after deployment, however the original version will be used at runtime until the newer version is deployed. To activate and apply the latest changes, you must redeploy the artifact.

You can use WLST commands to perform the following operations on the MDS Repository.

- *getSourceDeploymentHistory* used to list all the deployment history.
- *exportDeployedArtifact* used to export the deployed versions of the artifact.
- *deleteArtifactDeployment* used to delete the older deployment history.

For detailed information on using the MDS Repository, see *Managing the Metadata Repository* in *Administering Oracle Fusion Middleware*.

Interpreting Artifact Instance Messages

The Messages view is a searchable table of file delivery instance messages for the artifact for which the dashboard was opened. You can use % as a wildcard.

This table displays:

- Report: Click on the report link to open a source, transfer, or target report. See [Source Reports](#), [Transfer Reports](#), or [Target Reports](#).
- Purge: Click Purge to remove the selected source instance and related target and transfer instances. The source is not deleted but only the selected source instance and its related target and transfer instances are purged from the system.
- File Name: The name of the transferred file. The file might be renamed at the target; see [Moving and Renaming Files After Delivery](#).
- Size: The size in bytes of the transferred file. Compression or encryption might change the file size at the target; see [Setting Up Source Processing Actions](#) and [Setting Up Transfer Preprocessing and Postprocessing Actions](#).
- Create Time: The date and time that the file was written to the source or target location.
- Status: Active, Failed, or Completed. Substatuses of Active are:
 - Delivery In Progress
 - In Progress
 - Paused
 - Pending
 - Pending Delivery
 - Post Processing
 - Pre Processing
 - Retrying
 - ScheduledSubstatuses of Failed are:
 - Error
 - Partial Error
 - Post Processing
 - Pre ProcessingThe only substatus of Completed is Successful.
- Priority: Priority of the transfer.

Monitoring Transfer Flows Using the Main Dashboard

Use the main Dashboard to monitor the transfers. The main Dashboard displays real-time and recent file-delivery data. The Dashboard tab on the Monitoring page is always open.

The Main Dashboard displays data in four views:

- Metrics: See [Interpreting Main Dashboard Metrics](#).

- File Finder: See [Using the File Finder](#).
- Recent Errors: See [Diagnose Transfer Errors](#).
- Active Deliveries: See [Monitoring Active Deliveries](#).

You can expand each view to cover the entire tab by clicking the **Expand** icon in the upper right corner. To collapse the view and return to the Main Dashboard, click the **Restore** icon in the upper right corner of the expanded view.

The Design page and the Deployments tab show source, transfer, and target *artifacts*, which structure how file deliveries happen. In contrast, the Main Dashboard shows source, transfer, and target *instances*, which are the individual file deliveries themselves.

Interpreting Main Dashboard Metrics

The Metrics view displays a variety of statistics about recent transfers. You can use these statistics to determine which transfers are used frequently, failing, or taking a long time to complete.

To determine how recent the transfers are, select one of these interval values from the **View Data** drop-down list:

- Last Hour
- Last 6 Hours
- Last 24 Hours (default)
- Last 3 Days
- Last Week

Note:

If you use 'truncate table' to clean the environment, the materialized view will stop refreshing and metrics on MFT dashboard will not refresh. To fix this issue, you need to execute a database procedure `MFT_REFRESH_MV` which will enable refresh of materialized view again.

The statistics displayed in the unexpanded view are:

- Transfers: Total, Completed, Active, and Failed
- Most Active Transfers: Transfers with the most instances in the selected interval
- Status: A pie chart of completed, active, and failed transfers
- Failure Ratio: The percentage of failed transfers
- Payload File Size: Average, Minimum, and Maximum in bytes
- Transfer Speed: Average, Minimum, and Maximum in bytes per second
- Total Time: Average, Minimum, and Maximum in milliseconds

The expanded view adds an All Transfers table with these columns:

- Transfer: Click on the name to open a transfer report. See [Transfer Reports](#).
- Refresh: Refresh the page with updated data.

- Eraser Icon: Clears a selection from the transfer table. The icon is enabled if any section of the pie chart is clicked causes the transfer table to be displayed.
- Sort By: Choose whether to sort by status or by name.
- Source: The source associated with the transfer. Click on the name to open a source report. See [Source Reports](#).
- Status: Active, Failed, or Completed
- Substatus: The transfer substatus. The substatuses of Active are:
 - Delivery In Progress
 - In Progress
 - Paused
 - Pending
 - Pending Delivery
 - Post Processing
 - Pre Processing
 - Retrying
 - ScheduledSubstatuses of Failed are:
 - Error
 - Partial Error
 - Post Processing
 - Pre ProcessingThe only substatus of Completed is Successful.
- Initiation Time: The date and time that the transfer started
- Content Type: The type of the file transferred, for example, XML. Content filters define content types. See [Setting Up Content Filters](#).

Using the File Finder

The File Finder view provides a table that lets you search for transfer instances using names of associated files, sources, or targets as your search criteria.

In the unexpanded view, you can search for a file name, a source or target name, or both. Searches are case-sensitive and use the Starts With operator. You can use % as a wildcard. Table columns are:

- Source or Target: Click on the name to open a source or target report. See [Source Reports](#) or [Target Reports](#).
- Status: Active, Failed, or Completed
- Create Time: The date and time that the file was written to the source or target location
- File: The name of the transferred file. The file might be renamed at the target; see [Moving and Renaming Files After Delivery](#).
- Size: The size in bytes of the transferred file. Compression or encryption might change the file size at the target; see [Setting Up Source Processing Actions](#) and [Setting Up Transfer Preprocessing and Postprocessing Actions](#).

The expanded view provides additional features:

- A choice of operators: Starts With, Exact Match, or Contains
- A Case Sensitive check box
- An icon you can use to clear all search parameters
- Two additional table columns:
 - Type, with these possible values: B2B, SOA, OSB, ODI, Source Resubmit, or External
 - Sender, the name of the user that initiated the transfer
- Search fields for each column

 **Note:**

Postprocessing occurs after file delivery. Therefore, the Active Deliveries and File Finder views show different statuses if file delivery succeeds but postprocessing fails. Specifically, the Active Deliveries view displays a Completed status but the File Finder view displays a Failed status.

Monitoring Active Deliveries

The Active Deliveries view is the same unexpanded and expanded. For recent and in-progress deliveries, it displays:

- Transfer Name: Click on the name to open a transfer report. See [Transfer Reports](#).
- Start Time: The date and time that the transfer started.
- Progress: The percent complete. If the progress is less than 100% and is not changing, this might indicate that the transfer is paused or has failed. See [Pause and Restart a Transfer](#) or [Diagnose Transfer Errors](#).
- Refresh: Choose the refresh interval of active deliveries.

Interpreting Dashboards for All Transfers, Sources, or Targets

A source, transfer, or target dashboard displays real-time and recent data for file deliveries, or instances, that is related to all artifacts of the same type. Unlike the main Dashboard, an all-artifact dashboard must be opened.

Click **Transfer Instances**, **Source Instances**, or **Target Instances** in the left pane navigator to view the dashboard. The views on an artifact dashboard are:

- Messages: See [Interpreting Instance Messages](#).
- Summary Statistics: See [Interpreting Dashboard Metrics](#).

Unlike the views on the Main Dashboard, the views on the artifact dashboards have no **Expand** icons.

Interpreting Instance Messages

The Messages view is a searchable table of file delivery instance messages for all artifact of the type for which the dashboard was opened. You can use % as a wildcard. This table displays:

- ID: Click on the ID link to open a single-artifact source, transfer, or target dashboard. See [Interpreting Single Artifact Transfer_ Source_ and Target Dashboards](#).
- Name: The name of the artifact.
- Status: Active, Failed, or Completed. Substatuses of Active are:
 - Delivery In Progress
 - In Progress
 - Paused
 - Pending
 - Pending Delivery
 - Post Processing
 - Pre Processing
 - Retrying
 - Scheduled
 Substatuses of Failed are:
 - Error
 - Partial Error
 - Post Processing
 - Pre Processing
 The only substatus of Completed is Successful.
- URI: The URI for the source location, in the source dashboard only.
- Type: The source or target type, in the source and target dashboards only.
- Create Time: The date and time that the file was written to the source or target location.

Interpreting Dashboard Metrics

The Summary Statistics view displays a variety of statistics about recent transfers related to the artifact type for which the dashboard was opened. You can use these statistics to determine whether these transfers are used frequently, failing, or taking a long time to complete.

To determine how recent the transfers are, select one of these interval values from the **View Data** drop-down list:

- Last Hour
- Last 6 Hours
- Last 24 Hours (default)
- Last 3 Days
- Last Week

The statistics displayed are:

- Transfers: Total, Completed, Active, and Failed
- Status: A pie chart of completed, active, and failed transfers
- Most Active Artifacts: Names of the most active transfers, sources, or targets

- Failure Ratio: The percentage of failed transfers
- Payload File Size: Average, Minimum, and Maximum in bytes
- Transfer Speed: Average, Minimum, and Maximum in bytes per second
- Total Time: Average, Minimum, and Maximum in milliseconds

Interpreting Single Artifact Transfer, Source, and Target Dashboards

A single-artifact source, transfer, or target dashboard displays real-time and recent data for file deliveries, or instances, that is related to a specific artifact. Unlike the main Dashboard, an artifact dashboard must be opened.

The steps for this process are:

1. Click the arrow to the left of **Transfer Instances**, **Source Instances**, or **Target Instances** in the left pane navigator.

The artifacts of that type are listed.

2. Click the artifact name or right-click it and then select the Open menu item.

The dashboard tab for the artifact opens.

The views on an artifact dashboard are:

- Messages: See [Interpreting Artifact Instance Messages](#).
- Artifact Information: See [Interpreting Artifact Information](#).
- Summary Statistics: See [Interpreting Artifact Dashboard Metrics](#).

Unlike the views on the Main Dashboard, the views on the artifact dashboards have no **Expand** icons.

Interpreting Artifact Instance Messages

The Messages view is a searchable table of file delivery instance messages for the artifact for which the dashboard was opened. You can use % as a wildcard.

This table displays:

- Report: Click on the report link to open a source, transfer, or target report. See [Source Reports](#), [Transfer Reports](#), or [Target Reports](#).
- Purge: Click Purge to remove the selected source instance and related target and transfer instances. The source is not deleted but only the selected source instance and its related target and transfer instances are purged from the system.
- File Name: The name of the transferred file. The file might be renamed at the target; see [Moving and Renaming Files After Delivery](#).
- Size: The size in bytes of the transferred file. Compression or encryption might change the file size at the target; see [Setting Up Source Processing Actions](#) and [Setting Up Transfer Preprocessing and Postprocessing Actions](#).
- Create Time: The date and time that the file was written to the source or target location.
- Status: Active, Failed, or Completed. Substatuses of Active are:
 - Delivery In Progress

- In Progress
- Paused
- Pending
- Pending Delivery
- Post Processing
- Pre Processing
- Retrying
- Scheduled

Substatuses of Failed are:

- Error
- Partial Error
- Post Processing
- Pre Processing

The only substatus of Completed is Successful.

- Priority: Priority of the transfer.

Interpreting Artifact Information

The Artifact Information view displays the following information, if applicable, for the artifact for which the dashboard was opened. For sources, information about referencing transfers is also displayed. For transfers, information about the referenced source and targets is also displayed.

- Name: The artifact name.
- Category: The artifact category.
- URI: The endpoint location of a source or target.
- Binding Type: The source or target type. See [Source Types](#) and [Target Types](#).
- Deployed: Whether the artifact is deployed, `true` or `false`. See [Deploying a Source_ Target_ or Transfer](#).
- Label: The name of the deployed artifact on the Oracle WebLogic Server dedicated to Oracle Managed File Transfer.
- Status: Whether the deployed artifact is `Enabled` or `Disabled`. See [Disabling Sources_ Targets_ and Transfers](#) and [Reenabling Sources_ Targets_ and Transfers](#).
- Version: The artifact version.
- Create Time: The date and time that the artifact was created.

To open the artifact for which the dashboard was opened in a tab on the Designer page, click **View Definition**. See [Creating a Source](#), [Creating a Target](#), or [Configuring a Transfer](#).

Interpreting Artifact Dashboard Metrics

The Summary Statistics view displays a variety of statistics about recent transfers related to the artifact for which the dashboard was opened. You can use these statistics to determine whether these transfers are used frequently, failing, or taking a long time to complete.

To determine how recent the transfers are, select one of these interval values from the **View Data** drop-down list:

- Last Hour
- Last 6 Hours
- Last 24 Hours (default)
- Last 3 Days
- Last Week

The statistics displayed are:

- Transfers: Total, Completed, Active, and Failed
- Status: A pie chart of completed, active, and failed transfers
- Failure Ratio: The percentage of failed transfers
- Payload File Size: Average, Minimum, and Maximum in bytes
- Transfer Speed: Average, Minimum, and Maximum in bytes per second
- Total Time: Average, Minimum, and Maximum in milliseconds

Interpreting Source, Transfer, and Target Reports

A report provides a wealth of detailed information about the involvement of a source, transfer, or target in a specific file delivery instance.

You can access a report from:

- The Transfer or Source column of the All Transfers table in the expanded Metrics view. See [Interpreting Main Dashboard Metrics](#).
- The source or Target column in the File Finder view. See [Using the File Finder](#).
- The Transfer Name column in the Active Deliveries view. See [Monitoring Active Deliveries](#).
- The Report link in the Messages view of a source, transfer, or target dashboard. See [Interpreting Artifact Instance Messages](#).
- Another report. See [Using the Flow Diagram](#), [Source Reports](#), [Transfer Reports](#), and [Target Reports](#).

You can also perform operations on transfers from reports. See [Pause and Restart a Transfer](#) and [Resubmit a Transfer](#).

Using the Flow Diagram

At the top of every source, transfer, or target report is a flow diagram, which shows all artifacts related to the file delivery instance for which the report is generated. [Figure 5-1](#) shows a typical flow diagram.

Figure 5-1 Flow Diagram



The source is highlighted, indicating that this flow diagram is for a source report. You can click on the transfer or target icon to go to the transfer or target report. Each artifact has a green check, which indicates successful file delivery at each stage.

A down arrow to the right of an artifact indicates a fan-out. A flow diagram for a transfer with more than one target shows a fan-out at the target. [Figure 5-2](#) shows a fan-out at the transfer, which means that more than one transfer uses the same source.

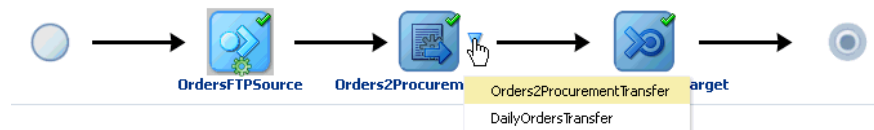
Figure 5-2 Flow Diagram with Fan-Out



The green gear on the source icon indicates that a preprocessing action, such as compression or encryption, is being performed at the source.

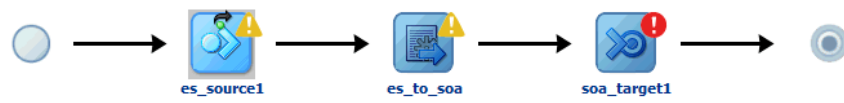
To display a list of all the artifacts in the fan-out, click the down arrow as shown in [Figure 5-3](#). You can then select from the list and go to the report for that artifact.

Figure 5-3 Flow Diagram with Fan-Out List



A flow diagram can also indicate warnings, errors, and resubmission attempts. [Figure 5-4](#) shows a flow diagram for a file delivery instance that was resubmitted at the source.

Figure 5-4 Flow Diagram with Resubmission, Warning, and Error Icons



The green circle with the black arrow around it on the source icon indicates that a resubmit occurred. The yellow triangles with exclamation points indicate that warnings occurred at the source and transfer. The red circle with the exclamation point indicates that an error occurred at the target.

Source Reports

A source report includes the following sections:

- Flow Display and Control: The flow diagram, Refresh button, report title, and Resubmit Source button

- **Summary:** The file name, source location, payload size, resubmission count, and other basic information
- **Error:** The error code, level, category, instance ID, timestamp, description, and other information about the error, if one occurred
- **Resubmitted Instances:** The instance ID, user comment, user name, timestamp, and other information about the source resubmission, if one occurred
- **Transfer Instances:** Information about transfers associated with the source. Click on a transfer name to go to the transfer report.
- **Source Processing:** Information about preprocessing actions such as compression, decompression, and encryption
- **Advanced:** Less commonly accessed information such as the source label and internal ID
- **Security:** Information related to security, if applicable, such as algorithms, certificates, and encryption

For full details about the information displayed in source reports, see *Source Report* in the *MFT Composer Online Help*.

Transfer Reports

A transfer report includes the following sections:

- **Flow Display and Control:** The flow diagram, Refresh button, report title, and Resubmit Transfer button
- **Summary:** The file name, status, resubmission count, and other basic information
- **Error:** The error code, level, category, instance ID, timestamp, description, and other information about the error, if one occurred
- **Resubmitted Instances:** The instance ID, user comment, user name, timestamp, and other information about the transfer resubmission, if one occurred
- **Target Delivery:** Information about targets associated with the transfer. Click on a target name to go to the target report.
- **Advanced:** Less commonly accessed information such as the internal ID
- **Security:** Information related to security, if applicable, such as algorithms, certificates, and encryption

For full details about the information displayed in transfer reports, see *Transfer Report* in the *MFT Composer Online Help*.

Target Reports

A target report includes the following sections:

- **Flow Display and Control:** The flow diagram, Refresh button, report title, Pause button, Resume button, and Resubmit Target button
- **Summary:** The file name, target location, total time, resubmission count, and other basic information
- **Error:** The error code, level, category, instance ID, timestamp, description, and other information about the error, if one occurred
- **Resubmitted Instances:** The instance ID, user comment, user name, timestamp, and other information about the target resubmission, if one occurred

- **Target Pre-Processing:** Information about preprocessing actions such as compression, decompression, and encryption
- **Target Post-Processing:** Information about postprocessing actions such as decompression
- **Advanced:** Less commonly accessed information such as the transfer and target labels and internal target ID
- **Security:** Information related to security, if applicable, such as algorithms, certificates, and encryption

For full details about the information displayed in target reports, see *Target Report* in the *MFT Composer Online Help*.

Pause and Restart a Transfer

You can pause an in-progress transfer from the target report, and restart a paused transfer.

You can access the target report in these ways:

- In the File Finder view, change **Search For** to Target, search for the file or target name, then click the name of the target to open the report.
- In the Active Deliveries view, click the name of the transfer to open the transfer report, then click the target icon in the flow diagram to open the target report.



Note:

If a transfer has multiple targets, you must pause the transfer for each target separately.

From the target report, click **Pause** to pause the file delivery to the target. To restart file delivery, click **Resume**.

Diagnose File Delivery Failures

Sometimes no error occurs, but a file does not arrive at the target location as expected. If this occurs, check the following settings:

- Content filters may exclude some files from delivery. See [Setting Up Content Filters](#).
- Files may be moved or renamed after delivery. Look for **Modified File Name** or **Target File Name** in the report. See [Source Reports](#), [Transfer Reports](#), or [Target Reports](#) and [Moving and Renaming Files After Delivery](#).
- File deliveries may occur only at specific dates and times. Look for a **Status** of Scheduled in the report. See [Source Reports](#), [Transfer Reports](#), or [Target Reports](#) and [Setting Up Schedules](#).

Resubmit a Transfer

You can resubmit a successful transfer to redeliver the file. You can resubmit it from the source, transfer, or target report.

Click the **Resubmit Source**, **Resubmit Transfer**, or **Resubmit Target** button. If a transfer has a status of Failed, you can resubmit it to try again. Whether you resubmit from the source,

transfer, or target report depends on where the failure occurred. For more information about errors, see [Diagnose Transfer Errors](#).

Resubmission occurs immediately and any schedule defined in the parent artifact is ignored. For example, if you resubmit from the target, any schedule defined at the target is ignored. However, if you resubmit from the source, a schedule defined at the target is still honored.



Note:

You need to manually refresh the page to view the details of the resubmit status.

The Resubmit button may be disabled for a number of reasons, including:

- The file to be transferred has been deleted.
- The transfer is still in progress.
- The transfer has a schedule and the delivery time is in the future.
- The target artifact has been disabled.
- You don't have permission to perform resubmit actions.
- You don't have permission to resubmit this particular transfer.

Bulk Resubmit

You can resubmit messages in bulk using an asynchronous WLST command. Multiple criteria provide you with the flexibility to customize the resubmissions to meet your needs.

For example, you could choose to resubmit all the failed messages to a given target for a given duration or resubmit the source for all the instance messages for a given time duration.

Supported Criteria

Syntax:

```
resubmitMessages (resubmitType, state, artifactName, startDate, endDate,
chunkSize, chunkDelay, ignoreIds, comments, previewMode)
```

You can choose from the following criteria for bulk resubmits:

Option	Description	Comments
<i>resubmitType</i>	SOURCE, TARGET, TRANSFER_INSTANCE, TARGET_INSTANCE	Required Example: wls:/mydomain/serverConfig> resubmitMessages('SOURCE')
<i>state</i>	FAILED, COMPLETED, ACTIVE	Optional. The system will consider all states (FAILED, COMPLETED, ACTIVE).
<i>artifactName</i>	The name of the artifact.	Optional. Example: wls:/mydomain/serverConfig> resubmitMessages('SOURCE','FAILED', artifactName='test src')

Option	Description	Comments
<i>startDate</i> <i>endDate</i>	Start and/or end date for message creation. Format: dd-MM-yyyy H:m:s:S	Optional. Example: wls:/mydomain/serverConfig> resubmitMessages('SOURCE','FAILED', startDate="12-12-2050 00:00:00:00", endDate="30-12-2050 00:00:00:00")
<i>chunkSize</i>	Batch size for resubmit.	Optional. Default = 1000 messages
<i>chunkDelay</i>	Delay in seconds between the two batch resubmits.	Optional Default = 30 seconds Example: wls:/mydomain/serverConfig> resubmitMessages('TRANSFER_INSTANCE','FAILED', artifactName='test transfer', chunkSize=100, chunkDelay=60, comments='test', previewMode='false')
<i>ignoreIds</i>	Comma separated list of corresponding message IDs to be ignored during resubmit.	Optional.
<i>comments</i>	Add any user comments.	Optional.
<i>previewMode</i>	When the command is run in preview mode, it lists the count of messages that will be resubmitted for the given criteria as well the entire WLST object.	Optional. Default = true If true, no record will be resubmitted. It can be used to get info such as how many records will be resubmitted for given criteria. Example: wls:/mydomain/serverConfig> resubmitMessages('TRANSFER_INSTANCE','FAILED', artifactName='test transfer', startDate="12-12-2050 00:00:00:00", endDate="30-12-2050 00:00:00:00", chunkSize=100, chunkDelay=60, ignoreIds='42F03A2A-D8DB-4EF5-A295-246EA93FAB29,5A6AA40C-61CC-4B5B-8E65-0D3D27098CD0', comments='test', previewMode='false')

Chunking and Resubmit Delay Duration Support

The criteria given for a resubmit may result in a huge number of instances to be resubmitted and a high load on the system. To avoid such scenarios, you can optionally specify two options: *chunkSize* and *chunkDelay*. For example, if the resubmit criteria results in 1000 records and chunk size is 200 and resubmit delay is 30 sec then 5 chunks(1000/200) are created and the delay between two chunks is 30 sec.

Based on the average expected time to process the resubmit message, you should provide the appropriate chunk size and the delay time by either specifying the parameter as part of the resubmit command, or by overriding the system defaults using the MBeans for these properties:

- `resubmitChunkSizeDefault`
- `chunkResubmitDelayDefault`

For more information about MBeans, see *Understanding WebLogic Server MBeans in Developing Custom Management Utilities Using JMX for Oracle WebLogic Server*.

Diagnose Transfer Errors

The unexpanded Recent Errors view on the Dashboard tab of the Monitoring page displays error codes, messages with endpoints, and timestamps.

The expanded Recent Errors view is a searchable table of errors, with additional information, including detailed descriptions, instance types (source, transfer, or target), levels, and severities. You can use % as a wildcard.

Clicking an error code in the unexpanded view or an error message in the expanded view opens the report for the source, transfer, or target at which the error occurred. The report includes an additional section with the error information. From the report, you can view additional information and resubmit the transfer; see [Resubmit a Transfer](#).

Diagnosing Error Messages and Descriptions

For more details about specific error messages that can appear in the MFT Console, such as in the Recent Errors view, on the WLS command line, or in the diagnostic logs, see *Error Messages*.

6

Administering Oracle Managed File Transfer

Learn how to administer the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer.

This chapter includes the following sections:

- [Changing Server Properties](#)
- [Importing and Exporting the MFT Configuration](#)
- [Increasing Memory to Improve Performance of Large File Transfers](#)
- [Oracle WebLogic Server Startup and Shutdown](#)
- [Transferring Files Through Firewalls Using the MFT FTP Proxy Server](#)
- [Managing Multiple Weblogic Servers and High Availability](#)
- [Enabling Event Notifications](#)
- [Configuring Oracle Managed File Transfer Error Processor Queues](#)
- [Oracle Managed File Transfer Loggers](#)
- [Viewing Oracle Managed File Transfer Log Messages](#)
- [Moving Oracle Managed File Transfer to Another Environment](#)
- [Managing Keystores Using Oracle MFT Console](#)
- [Adding Purge Schedules](#)

For information about keystores, see [Managing Keystores](#).

For information about domains, see [Managing Domains](#).

For information about administering Oracle Managed File Transfer embedded FTP and sFTP servers, see [Administering Oracle Managed File Transfer Embedded Servers](#).

Changing Server Properties

You can change or update server properties using the Server Properties tab on the Administration page.

The Server Properties tab on the Administration page is arranged in the following sections:

- [General Server Configuration Properties](#)
- [Performance Properties](#)
- [High Availability Properties](#)
- [Advanced Delivery Properties](#)
- [Runtime MBean Properties](#)

To save changes to these properties, click **Save**. To cancel all changes since you last saved, click **Revert**.

General Server Configuration Properties

The **Server Properties** tab displays the following configurable fields:

- **Payload Storage Directory:** The directory location where the MFT payloads are stored. This has to be a shared location. The default **Payload Storage Directory** is `WLS_Home/user_projects/domains/base_domain/mft/storage`. You must set it to a shared location if multiple Oracle WebLogic Server instances run in a cluster.
- **Callout Directory:** The shared directory location where custom callout JARs are stored. MFT loads the callout jars from this location. The callout processing will fail if no required jar is present in the given location. The default location is `WLS_Home/user_projects/domains/base_domain/mft/callouts`.
- **Store Inline Payload:** The **Store Inline Payload** setting determines whether inline payloads are stored in the File System or a Database after transfer. You must select one of the options.

If you select File System, the **Payload Storage Directory** specifies the File System location. If the configured inline payload size is more, then File System is recommended.

If you select Database, the payload is stored in the database that stores the Oracle Managed File Transfer configuration.

- **Store Reference Payload:** The **Store Reference Payload** checkbox determines whether referenced payloads are stored in the payload storage directory. If you check this box, the **Payload Storage Directory** specifies the file system location, if you do not select the checkbox, then no reference payload will be persisted.

To change the **Payload Storage Directory**:

1. Stop the Oracle WebLogic Server managed server(s) dedicated to Oracle Managed File Transfer.
2. Change the **Payload Storage Directory** setting.
3. Move the directories and files under the **Payload Storage Directory**.
4. Restart the managed server(s).

See [Re-configuring the Port](#) for how to stop and start the managed server(s).

- **Generate Checksum:** The **Generate Checksum** setting helps you verify that all the bits in a file were successfully transferred. If this box is checked, Oracle Managed File Transfer generates a checksum for every payload before delivering it to target and stores it in the database. This checksum can then be compared to verify that the file was not corrupted during transfer.

Note:

Checksum is necessary to validate the payload in the later stage. This is an optional field, if you do not need MFT to generate checksum, then do not select this field. This will avoid additional checksum computation thus increasing the performance.

Performance Properties

The Processors settings determine the number of processing threads dedicated to each stage of file delivery. The default for all three settings is 2.

- **Source Processors:** determines the number of message processors required at the source level. This handles transfer identification and source-level processing function execution.
- **Instance Processors:** determines the number of message processors required at the transfer level. This handles target-level preprocessing function execution.
- **Target Processors:** determines the number of message processors required at the target level. This handles delivery and target-level postprocessing function execution.

In most cases, this number has to be more because much of load like target message processing or delivery happens in this layer. So usually this can be higher than other two types processors.

Based on the transfer configuration, processing functions defined, and expected payload pattern, you might want to increase or decrease the number of threads assigned to each Processors setting. For example, if the time taken for target preprocessing is short compared to delivery, you might want to reduce the number of **Instance Processors** and increase the number of **Target Processors** for optimal performance.

Oracle Managed File Transfer uses the collision detection feature of the internal JCA adapter to prevent thread concurrency issues.

High Availability Properties

These settings are JCA adapter properties for high availability:

- **Control Directory** is the directory path which MFT File/FTP adapters require to handle HA use cases. This field is required if the MFT is running in HA environment. You must set it to a shared location if multiple Oracle WebLogic Server instances run in a cluster, for example `$DOMAIN_HOME/mft/control_di`.
- **Inbound Datasource** is the inbound data source of MFT where the schemas corresponding to high availability are precreated. This field is required if the Control Directory is not provided. The default, established outside of Oracle Managed File Transfer, is `jdbc/MFTDataSource`.
- **Outbound Datasource** is the outbound data source of MFT where the schemas corresponding to high availability are precreated. This field is required if the Control Directory is not provided. The default, established outside of Oracle Managed File Transfer, is `jdbc/MFTDataSource`.

For more information, see *Configuring Oracle File and FTP Adapters for High Availability in Understanding Technology Adapters*.

Advanced Delivery Properties

The Advanced Delivery settings are required when Oracle WebLogic Server instances are run with a load balancer. These settings capture the **Internal Address** and **External Address** (IP addresses) and the **FTP**, **FTPS**, and **sFTP** ports that the load balancer uses. Use this setting when Oracle Managed File Transfer sends a payload as an FTP or sFTP reference. If the values are set, they are used to construct the FTP reference (FTP/sFTP host address and ports).

Internal Address: Internal proxy address and ports needed for reference delivery cases.

External Address: External proxy address and ports needed for reference delivery cases.

If Oracle MFT is running behind internal and external proxies, then the Internal and External IP addresses are required.

For more information, see Load Balancing in a Cluster in *Administering Clusters for Oracle WebLogic Server*.

Runtime MBean Properties

You can add or update MBean properties using the Enterprise Manager Console.

To add or update MBean properties:

1. Log in to the Enterprise Manager Console.
2. Click **Target Navigation**.
3. Under **MFT**, go to **mft-app(server_name)**.
4. At the bottom, click **More MFT Configuration Properties**.
5. Click the **Operations** tab and add or update properties as required:
 - To add a new property, use `addProperty`. Specify the name and value of the MBean, as described in the table below.
 - To change the value of a property, use `setProperty`.
 - To get the value of a property, use `getProperty`.

For example:

```
addProperty: Name: purgeTransactionTimeout Value: 3600
```

Property	Description	Value Unit	Default Value
<code>taskThreadPoolSize</code>	The non-JCA download thread count (shared by all non-JCA sources).	integer	5
<code>sftpFileLastModifiedInterval</code>	The interval at which the file last modified date will be updated to the underlying file system when uploading the files to the SFTP server. This is used to reduce the number of modified date calls to the DBFS file system in the MFT Cloud Service environment that was causing the SFTP performance issue in the cloud environment.	seconds	-1

Property	Description	Value Unit	Default Value
sftpAlgorithms	<p>Security algorithms to be supported for the MFT SFTP embedded server. This can be used to exclude support for the weaker or unwanted algorithms from the server. This property supports the following algorithms:</p> <ul style="list-style-type: none"> • KE (KeyExchange) • MA (MessageAuthentication) • SG (Signature) • CP (Ciphers) <p>Example: KE=diffie-hellman-group1-sha1,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha256:MA=hmac-md5,hmac-sha1,hmac-md5-96,hmac-sha1-96,hmac-sha2-256,hmac-sha2-512:SG=ssh-dss,ssh-rsa:CP=aes128-cbc,aes192-cbc,aes256-cbc,3des-cbc,blowfish-cbc,aes128-ctr,aes192-ctr,aes256-ctr,arcfour128,arcfour256</p>	string	N/A
releaseUnprocessedLocks	If there is a connection issue with getting file information (post listing), the adapter does not release the lock immediately, and recovery to clean up stuck locks begins only after 10 (non-zero files) connection attempts. When set to <code>true</code> , locks are released immediately.	boolean	true
releaseLockCycle	If there is a connection issue with getting file information (post listing), the adapter does not release the lock immediately, and recovery to clean up stuck locks begins only after 10 (non-zero files) connection attempts. This property specifies the number of lock cycles after which the lock should be released.	integer	1
purgeTransactionTimeOut	The timeout within which the purge transaction should complete. The recommended setting is a value based on the volume or number of files in the file system.	seconds	-1

Property	Description	Value Unit	Default Value
progressMonitorTimeToCommit	The time lapse between commits by the progress monitor thread. This is used to show the progress bar for an ongoing transfer. The data will be refreshed only after this time lapse.	milliseconds	4000
pgpSigningKeyPassphrase	The passphrase for PGP signing.	string	empty
pgpFailOnUnknownSignature	When set to <code>true</code> , generates an error for an unknown signee for the PGP payload.	boolean	false
pendingMessagesWaitTimeMillis	The wait time to facilitate the transfer of large files.	milliseconds	3600000 (one hour)
outboundMutexType	<code>outboundMutexType" = (oracle:jdbc/SOADDataSource OR oracle:jdbc/MFTDataSource OR coherence:jdbc/MFTDataSource, and so on). This is to have a unique SEQ number in the case of a cluster.</code>	string	empty
notificationCalloutTimeout	The global error notification callout timeout.	seconds	30
ModificationTimeFormat ModificationTimeFormatOld	The modification time format. (FTP seeks two types of formats, hence two MBeans are required to facilitate both)	string Format: <i>integer</i> , <i>integer</i> , <i>date-time format</i>	4,18,yyyyMMddHHmmSS
minFileSizeForProgressMonitor	The minimum file size for which progress should be monitored. Not needed for small payloads or files.	MB	10
messageDigestAlgorithmForChecksum	The digest algorithm used for checksum calculation.	string	SHA256
maxTargetHttpTimeout	The connection timeout for a SOAP type source and target.	milliseconds	600000
maxMessageTemplateCacheCount	The file-based template cache count, used for successful transfer notification.	integer	20
KeystoreAlgorithm	The key protection algorithm used in the security store.	string	SunX509
inlinePayloadPrefix	The file name prefix when the incoming payload is inline and no target file name is available.	string	mft-
ignorePgpSigningValidation	When set to <code>true</code> , specifies to not generate an error when the PGP signature is not valid.	boolean	false
ignoreInitialFileWait	When set to <code>true</code> , specifies to ignore the wait time for the status to become true for the endpoint (for events).	boolean	false

Property	Description	Value Unit	Default Value
fsPurgeBatchSize	The maximum number of files to purge per single file system purge action. The recommended setting is a value based on the volume or number of files in the file system.	integer	5000
eventLastModifiedTimeLapse	The time lapse after which the event would be forced to be complete if there is no progress on the event or modification in the events tables.	seconds	3600 (one hour)
ESComprehensiveLogin	By default, the FTP and SFTP embedded servers sessions in MFT do not have login capabilities. Set this property to <code>true</code> to turn on the user lock, user validity, and other such login checks for FTP and SFTP sessions.	boolean	false
enableSingleInstanceJCAPolling	When set to <code>true</code> , enables single instance JCA polling in a multinode environment.	boolean	false
enableFIPSMODE	When set to <code>true</code> , enables FIPS 140 mode, which enforces more security for MFT.	boolean	false
enableFileCheck	For some FTP servers such as AS400, the list content is not parsable. When this property is set to <code>false</code> , FTP commands for non-read only polling are ignored. For read only polling, this is not required as you can use NLST.	boolean	true
enableDynamicTargetFoldername	When set to <code>true</code> , enables override of the target folder by providing the target folder via callout or via SOAP header.	boolean	false
enableAutoGeneratedDocName	When set to <code>true</code> , generates a document name in WebCenter.	boolean	true
disableTransferPriority	When set to <code>true</code> , disables the transfer level priority so that all messages will be processed with same priority.	boolean	false
disableExternalContacts	When set to <code>true</code> , disables the creation of any external contacts.	boolean	false
disableEventServices	When set to <code>true</code> , disables the MFT event service.	boolean	false
defaultFromEmailAddressForNotification	The default from email address for sending email notifications from MFT.	string	user@host.com

Importing and Exporting the MFT Configuration

A backup of the Oracle Managed File Transfer configuration (or repository) includes all Administration page settings and all Designer artifacts, excluding password settings. The configuration is saved to a ZIP file, which you can restore later.

The steps for this process are:

1. Open the Import/Export tab on the Administration page.
2. Click **Export**.

An operating system dialog box opens asking what action to take with the file `export.zip`.

3. Select **Save File** and click **OK**.

An operating system file saving dialog box opens.

4. Select the directory to which to save the file.
5. Edit the file name. This is optional.
6. Click **Save**.

You can restore a backup of the Oracle Managed File Transfer configuration. This overwrites the existing configuration except for password settings.

To import or export a single transfer artifact, see [Importing and Exporting Transfers](#).

You can use the `exportMftMetadata` and `importMFTMetadata` WLST commands to back up and restore the Oracle Managed File Transfer configuration. For more information, see *MFT Metadata Commands in WLST Command Reference for SOA Suite*.

Keystores are not part of the Oracle Managed File Transfer configuration. For more information, see [Managing Keystores](#).

Increasing Memory to Improve Performance of Large File Transfers

If transfers of large files are running slowly or out-of-memory exceptions occur, increase the memory (`-Xms`) or heap size (`-Xmx`) to 1 or 2 GB.

For example, the following command increases the memory to 1 GB and the heap size to 2 GB:

```
setenv USER_MEM_ARGS "-Xms1024m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=768m"
```

Oracle WebLogic Server Startup and Shutdown

You can start, restart, or stop the Oracle WebLogic Server managed server(s) dedicated to Oracle Managed File Transfer from the Oracle Managed File Transfer console.

Embedded servers are not the same thing as Oracle WebLogic Server managed servers. The MFT embedded FTP and sFTP servers are services that run on the Oracle WebLogic Server managed server(s) dedicated to Oracle Managed File Transfer. The MFT console page for embedded server ports management lists these managed servers and allows you to stop and start them.

The steps for this process are:

1. On the left pane of the Administration page, click the arrow to the left of **Embedded Servers**.

The **Ports** and **User Access** items appear.

2. Click **Ports**.

The Embedded Server Ports tab opens.

3. Select the **Server Instance** that you want to start, restart, or stop by checking its box in the table. To perform an operation on all servers, check the **Select All** box.
4. Look at the **Server Status**. You can stop or restart the server if it is RUNNING. You can start the server if it is STOPPED.
5. Click **Start**, **Stop**, or **Restart**.
6. Click the **Refresh** icon to update the table and verify the new **Server Status**.

The primary purpose of the **Ports** tab is updating embedded server ports. For details, see [Re-configuring the Port](#).

To start and stop only the embedded FTP and sFTP servers, see [Starting and Stopping Embedded Servers](#).

For more information about how to manage Oracle WebLogic Server startup, shutdown, and failure recovery, see Starting and Stopping Servers in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

Transferring Files Through Firewalls Using the MFT FTP Proxy Server

Oracle Managed File Transfer includes the MFT FTP Proxy Server that supports the FTP, sFTP and FTPS protocols.

Note:

MFT Proxy is a standalone Java application and it does not support HA or clusters. The MFT Proxy Server is typically used to connect to a single hardware load balancer, which in turn communicates with MFT Managed Servers in a cluster residing inside the firewall. It is used in a development environment or where HA is not required. Oracle recommends not using MFT Proxy in a production environment. It may be used for internal testing purposes.

The MFT FTP Proxy server has the following characteristics:

- Supports the FTP, sFTP, and FTPS protocols.
- Resides separately outside a firewall, eliminating the need to deploy MFT outside the firewall.
- Accepts external FTP requests and forwards them to MFT servers inside the firewall.
- Typically connects to a single hardware load balancer.
- Supports standard gateway and reverse proxy use cases.
- Allows for configuration of ports and inbound server IP addresses.
- Supports the use of any standard FTP client.

The following is a simple topology:

FTP Client > MFT FTP Proxy Server > Hardware Load Balancer > MFT Server Cluster

The files that comprise the MFT FTP Proxy Server are located in `WLS_HOME/mft/applications/proxy/config`. The `Readme.txt` file describes how to configure and deploy the MFT FTP Proxy Server.

For more information about setting up firewalls, see Security Options for Cluster Architectures in *Administering Clusters for Oracle WebLogic Server*.

Remote SFTP with Proxy Server

When Transport Provider is set as Socket, connection to remote server is direct, it does not use the proxy definition. When Transport Provider is set as HTTP, the connection to remote SFTP server is through a proxy server. If the proxy details are not entered or is invalid, then delivery will fail. When Transport Provider is set as HTTP and proxy settings are defined and are valid, MFT will deliver the file to target via proxy server.

Managing Multiple Weblogic Servers and High Availability

Oracle Managed File Transfer runs on Oracle WebLogic Server. Therefore, setting up high availability for Oracle Managed File Transfer depends on setting up high availability for Oracle WebLogic Server.

This section includes the following topics:

- [Configuring High Availability](#)
- [Preventing Cluster Startup Errors](#)
- [Load Balancing in Oracle Managed File Transfer](#)

Configuring High Availability

The high-level steps for this process are:

1. Set up the JCA adapter properties for high availability during Oracle WebLogic Server configuration. You need these for setting [High Availability Properties](#). For more information, see Configuring Oracle File and FTP Adapters for High Availability in *Understanding Technology Adapters*.
2. Create a cluster of managed servers during the domain configuration step of Oracle Managed File Transfer installation. For more information, see Configuring a Cluster for Oracle Managed File Transfer in *Installing and Configuring Managed File Transfer*.
3. Install a software or hardware load balancer. Note the **Internal Address** and **External Address** (IP addresses) and the **FTP**, **FTPS**, and **sFTP** ports that the load balancer uses. You need these for setting [Advanced Delivery Properties](#). For more information, see the load balancer documentation.
4. Configure [High Availability Properties](#) and [Advanced Delivery Properties](#) in Oracle Managed File Transfer and **Save** them.
5. Restart each managed server. See [Oracle WebLogic Server Startup and Shutdown](#).

For more information about topics such as cluster topologies, virtual IP addresses, load balancing options, failover, and so on, see Understanding WebLogic Server Clustering, Virtual Server IPs and Pool, and Load Balancing in a Cluster in *Administering Clusters for Oracle WebLogic Server*.

Preventing Cluster Startup Errors

If the Oracle WebLogic Server cluster uses unicast messaging, and the servers in the cluster don't synchronize right away, you might see a message such as the following when the cluster restarts:

```
<Error> <oracle.soa.adapter.jms.inbound> <BEA-000000>  
<JMSSMessageConsumer_init:[destination =  
MFTJMSServer_auto_2@jms/mft/MFTSourceQueue  
(payload = 1)]:ERRJMS_ERR_CR_QUEUE_CONS.  
Unable to create Queue receiver due to JMSEException.
```

You might see messages such as the following in a stack trace:

```
weblogic.jms.common.JMSEException: could not find Server mft_server1  
  
javax.naming.NameNotFoundException: Unable to resolve  
'weblogic.messaging.dispatcher.S:mft_server1'. Resolved  
'weblogic.messaging.dispatcher'; remaining name 'S:mft_server1'
```

You can usually ignore these messages, because server synchronization is retried and usually succeeds. If you prefer not to see this message, do one of the following:

- Set the Member Warmup Timeout Seconds MBean attribute to 30. This delays synchronization, giving the servers in the cluster time to start up first.
- Use multicast messaging, which sets the default of Member Warmup Timeout Seconds to 30.

For more information, see "Communications In a Cluster" in *Administering Clusters for Oracle WebLogic Server*.

Load Balancing in Oracle Managed File Transfer

Oracle Managed File Transfer has mechanisms to ensure that in a cluster, only one Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer executes a particular file transfer instance. Load balancing of transfers is sticky.

If a managed server fails while a transfer is in progress, the transfer is automatically resubmitted when that managed server restarts.

Enabling Event Notifications

You can notify specific users about Oracle Managed File Transfer events such as errors or new artifact deployments. Oracle MFT uses UMS for notifications.

To notify specific users about events, you configure notifications using WLST commands for Oracle Managed File Transfer, then you configure the email or SMS driver. Notifications event types are as follows:

- `RUNTIME_ERROR_EVENT` — Errors during events such as message processing, server start-up, source error, or system event failure error.
- `DELETE_ARTIFACT_EVENT` — Deletion of a transfer, source, or target.
- `DEPLOY_ARTIFACT_EVENT` — Deployment of a transfer, source, or target.
- `EXPORT_IMPORT_EVENT` — Import or export of the MFT configuration.

- `PURGE_EVENT` — Purging of runtime transfer instances or transfer payloads.
- `ARCHIVE_RESTORE_EVENT` — Archiving or restoring of runtime transfer instances or transfer payloads.

All notification messages are sent to JMS `MFTExceptionQueue`, whether or not they are enabled or there are contacts to be notified.. For more information about `MFTExceptionQueue`, see [MFTExceptionQueue](#).

The steps for enabling event notifications are:

By default all the event notifications are disabled. Use the following WLST commands to enable the MFT Notifications. For information on WLST commands, see [Running WLST Commands](#).

1. Start WLST:

```
$MW_HOME/mft/common/bin/wlst.sh
```

2. Connect to MBean server:

```
connect("adminusername","adminpassword","t3://localhost:mft-port")
```

where `mft-port` is the configured port of the MFT server.

3. Create users, or contacts, to notify. For example: `createContact('Email', 'jane.doe@example.com')`
4. Associate contacts with the notification events. For example: `addContactToNotification('DEPLOY_ARTIFACT_EVENT', 'Email', 'jane.doe@example.com')`
5. Enable the notification event. For example: `updateEvent('RUNTIME_ERROR_EVENT', true)`

The notification is enabled for the run-time message processing errors. Any runtime message error will be notified via email to specified email address.

For notifications of type `EMAIL` or `SMS` to work, you must configure an email or SMS driver. See [Configuring an Email Driver for Notifications](#) or [Configuring an SMS Driver for Notifications](#) for more information.

For more information about WLST notification commands for Oracle Managed File Transfer, see MFT Event Notification Commands in *WLST Command Reference for SOA Suite*.

Configuring an Email Driver for Notifications

Use Oracle Enterprise Manager Fusion Middleware Control to configure the email driver for event notifications.

Note:

By default all User Messaging Service (UMS) channels are enabled with the Oracle specific-settings (mail server). External customer need to change these settings in the Enterprise Manager for notification to work.

The steps for this process are:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the **User Messaging Service** node.

3. Expand the **usermessagingdriver-email** node.
4. Select the node for the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer. For example, this node might be named **usermessagingdriver-email (mft_server1)**.
5. In the **usermessagingdriver-email** page, select **User Messaging Email Driver > Email Driver Properties** to open the Email Driver Properties page.
6. On the Email Driver Properties page, enter values for your mail server for the following (mandatory) Outgoing and (optional) Incoming Mail Server properties:
 - Outgoing Mail Server
 - Outgoing Mail Server Port
 - Incoming Mail Server
 - Incoming Mail Server Port
 - Incoming Mail IDs
 - Incoming User IDs
 - Incoming Passwords
7. Click **Apply**.
8. Log out and log back in to the Fusion Middleware Control console and verify that these properties are saved correctly.
9. Restart the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer if necessary. See [Oracle WebLogic Server Startup and Shutdown](#).

The current `defaultFromEmailAddressForNotification` Mbean property to change the default sender address used for notifications is `no.reply@oracle.com`.

For more information about configuring the email driver in Fusion Middleware Control, see *Configuring the E-Mail Driver in Administering Oracle User Messaging Service*.

Configuring an SMS Driver for Notifications

Use Oracle Enterprise Manager Fusion Middleware Control to configure the SMS (or SMPP) driver for event notifications.

Note:

By default all User Messaging Service (UMS) channels are enabled with the Oracle specific-settings (mail server). External customer need to change these settings in the Enterprise Manager for notification to work.

The steps for this process are:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the **User Messaging Service** node.
3. Expand the **usermessagingdriver-smpp** node.
4. Select the node for the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer. For example, this node might be named **usermessagingdriver-smpp (mft_server1)**.

5. In the `usermessagingdriver-smpp` page, select **User Messaging SMPP Driver > SMPP Driver Properties** to open the SMPP Driver Properties page.
6. On the SMPP Driver Properties page, click **Create**.
7. On the Create SMPP Driver Properties page, enter values for the following properties:
 - SmsAccountId
 - SmsServerHost
 - TransmitterSystemId
 - ReceiverSystemId
 - TransmitterSystemType
 - ReceiverSystemType
 - DefaultSenderAddress
 - ServerTransmitterPort
 - TransmitterSystemPassword
8. Click **OK**.
9. Log out and log back in to the Fusion Middleware Control console and verify that these properties are saved correctly.
10. Restart the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer if necessary. See [Oracle WebLogic Server Startup and Shutdown](#).

For more information about configuring the SMS driver in Fusion Middleware Control, see *Configuring the SMPP Driver* in *Administering Oracle User Messaging Service*.

MFTExceptionQueue

When you install MFT, the MFT installer creates a JMS queue called MFTExceptionQueue. This queue is used to trace events for notifications that are not configured.

All notification messages are sent to JMS MFTExceptionQueue, whether or not they are enabled or there are contacts to be notified.

By using JMS MFTExceptionQueue, you can track the following events:

Table 6-1 Events

Events	Description
RUNTIME_ERROR_EVENT	This event tracks the errors occurred at runtime. For example: <ul style="list-style-type: none"> • Any errors that occur during the message processing • Server start errors • System event failure errors
EXPORT_IMPORT_EVENT	This event tracks both export and import events. If notification is not configured, MFT submits a message in exception queue. For example, a sample message has a whole repository export. This message is submitted only when an event is successful.
DEPLOY_ARTIFACT_EVENT	This event tracks the artifact deployment events. If a notification is not configured, MFT submits a message in exception queue. For example, a sample message indicates a deployment event for a source, for example, "osbsrc1". This message is submitted only if the deployment is successful.

Table 6-1 (Cont.) Events

Events	Description
DELETE_ARTIFACT_EVENT	This event tracks tip metadata deletion. For example, source, transfer and target deletion.
PURGE_EVENT	This event tracks the runtime/payload purge. MFT submits a message in exception queue when runtime/payload purge is executed. This message is submitted on successful completion of purge only if notifications are not configured. For example, a sample message shows a runtime purge.
ARCHIVE_RESTORE_EVENT	This event tracks the runtime / payload archive and restore. MFT submits a message in exception queue when archive / restore is executed for runtime/payload data. This message is submitted on successful completion of event only if notifications are not configured. For example, a sample message shows a runtime archive event.

Configuring Oracle Managed File Transfer Error Processor Queues

You can configure JMS queues to receive Oracle MFT runtime error events using the Oracle WebLogic Server Administration Console.

The preexisting MFT runtime processor queues are:

- MFTSourceQueue
- MFTInstanceQueue
- MFTTargetQueue

This section describes how to create corresponding error queues for each of these runtime processor queues.

In MFT, messages are processed asynchronously. A single MFT message is processed by three different processors and for each of these processors, there is a corresponding queue as listed below:

Table 6-2 Processors and Queues

Processors	Queues
Source Processors	MFTSourceQueue
Instance Processors	MFTInstanceQueue
Target Processors	MFTTargetQueue

Also, every MFT message corresponds to a JMS message for each of these processors. There can be exceptions to any of the error processor queues.

The expected and unexpected system-level errors are handled in the following way:

- Expected system-level errors: For every expected system-level error such as file not found there is a corresponding MFT message which is marked as "Error".

For delivery related expected errors such as remote server down, the corresponding message is retried based on the retry settings in the target. For other expected errors, there are no retries.

- **Unexpected system-level errors:** If there are unexpected system-level errors such as rac failures or JMS failures, MFT retries the JMS message three times. If all retries are exhausted, then the MFT messages are stuck in active status. The corresponding JMS message is purged or not depends on whether the JMS processor error queues are associated with MFT processing queues.

By default, the error processing queues are not configured. If this is the case, the JMS messages are purged otherwise the JMS messages are redirected to configured error queue.

If transfer instances are stuck in active status for a long time, you can check these error queues. The JMS messages in the error queues have the complete details of the transfer instances. You can search for the message ID in the MFT log files to find the root cause (error stack). Once the issue is resolved, you can re-send the JMS message to corresponding runtime queue for reprocessing.

The steps for creating error queues are:

1. Access the Oracle WebLogic Server console using a URL that includes the Oracle WebLogic Server hostname and the console port:

```
http://wls-hostname:console-port/console
```

For example:

```
http://localhost:7011/console
```

2. Log in using the Oracle WebLogic Server admin username and password.
3. In the left pane of the Oracle WebLogic Server Administration Console, expand the **Services** node and the **Messaging** node under it.
4. Select **JMS Modules**.

The Summary of JMS Modules page appears.

5. Select **MFTJMSModule** from the list of JMS modules.

The Settings for MFTJMSModule page appear.

Note that MFTSourceQueue, MFTInstanceQueue, and MFTTargetQueue are in the Summary of Resources list.

6. Click **New**.

The Create a New JMS System Module Resource page appears.

7. Select **Queue** and click **Next**.

8. Type a **Name** and a **JNDI Name** for the new queue and click **Next**.

For example, you could type `MFTSourceErrorQueue` for the Name and `jms/mft/MFTSourceErrorQueue` for the JNDI Name.

9. Select MFTSubDeployment from the **Subdeployments** drop-down list.

10. Select MFTJMSServer from the **Targets** list.

11. Click **Finish**.

The Settings for MFTJMSModule page reappear with the new queue added to the Summary of Resources list.

12. Select **MFTSourceQueue** from the Summary of Resources list.

The Settings for MFTSourceQueue page appear.

13. Select the **Delivery Failure** tab.
14. Select **Redirect** from the **Expiration Policy** drop-down list.
15. Select the new queue from the **Error Destination** drop-down list.
16. Click **Save**.
17. Select **MFTJMSModule** from the breadcrumb list at the top of the page.

The Settings for MFTJMSModule page reappear.

18. Repeat steps 6 through 17 to configure error queues for the MFTInstanceQueue and MFTTargetQueue JMS queues.

Configuring Oracle Managed File Transfer Loggers

You can configure the loggers for MFT components and MFT embedded servers.

To configure MFT component loggers, you use Oracle Enterprise Manager Fusion Middleware Control. To configure MFT embedded server loggers, you edit the configuration file directly.

MFT Component Logger Configuration

MFT component log messages are written to the log file at this location:

```
DOMAIN_HOME/servers/mft_server_name/mft_server_name-mft-diagnostic.log
```

Use Oracle Enterprise Manager Fusion Middleware Control to modify the sizes of the log files and to set the log level for the various loggers. The steps for this process are:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Weblogic Domain node.
3. Expand the node for the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.

For example, the domain might be `soainfra` or `base_domain`.
4. Right-click on the MFT server.

For example, the MFT server might be `mft_server1`.
5. Select **Logs > Log Configuration**.
6. Select the **Log Files** tab.
7. Select `odl-handler` and click **Edit Configuration**.
8. Edit the **Maximum Log File Size (MB)** and **Maximum Size of All Log Files (MB)** values and click **OK**.
9. Select the **Log Levels** tab.
10. Expand the **Root Logger** node and any other nodes necessary to display the loggers for which to set log levels.

Table 6-3 lists the loggers relevant to MFT components.

11. For each logger, select the desired **Oracle Diagnostic Logging Level (Java Level)**: `SEVERE+100`, `SEVERE`, `WARNING`, `INFO`, `CONFIG`, `FINE`, `FINER`, or `FINEST`. The default is `INFO`.
12. Check **Persist log level state across component restarts** if necessary.

13. Click **Apply**.

MFT Embedded Server Logger Configuration

MFT embedded server log messages are written to the log file at this location:

`DOMAIN_HOME/servers/mft_server_name/mft-es/mft_server_name-mft-es-diagnostic.log`

To modify the sizes of the log files and to set the log level for the various loggers, open the configuration file in a text editor to and make the required changes. The configuration file is at this location:

`DOMAIN_HOME/config/fmwconfig/servers/mft_server_name/mftes-log4j2.xml`

The following table lists the loggers relevant to MFT embedded servers.

Logger Name	Logged Events	Default Level	Production Level
org.apache.ftpserv	FTP operations	INFO	SEVERE
org.apache.sshd	sFTP operations	INFO	SEVERE

The configuration file changes take effect immediately. No server restart is required.

Oracle Managed File Transfer Component Loggers

Use this table as a reference for MFT component logger names, logged events, and their default levels.

Table 6-3 MFT Component Loggers

Logger Name	Logged Events	Default Level	Production Level
oracle.mft.ENGINE	Message processing, processing functions and callouts, deployment operations	INFO	SEVERE
oracle.mft.TRANSPORT	Transport layer interaction operations, system events	INFO	SEVERE
oracle.mft.COMMON	Audit, notification, purge, archive, restore, server startup, common utility APIs	INFO	SEVERE
oracle.mft.REPOSITORY	Java persistence operations	INFO	SEVERE
oracle.mft.METADATA	WLST and EJB invoked API handling	INFO	SEVERE
oracle.mft.SCHEDULER	Schedule operations	INFO	SEVERE
oracle.mft.EMBEDDED_SERVER	Embedded server communication with the MFT server	INFO	SEVERE

Table 6-3 (Cont.) MFT Component Loggers

Logger Name	Logged Events	Default Level	Production Level
oracle.mft.SECURITY	Security operations, OWSM local policy	INFO	SEVERE
oracle.mft.CONSOLE_UI	Console operations	INFO	SEVERE
oracle.mft.adapter.file	Not Applicable	INFO	SEVERE
oracle.mft.adapter.file.inbound	Not Applicable	INFO	SEVERE
oracle.mft.adapter.file.outbound	Not Applicable	INFO	SEVERE
oracle.mft.adapter.file.connection	Not Applicable	INFO	SEVERE
oracle.mft.adapter.file.transaction	Not Applicable	INFO	SEVERE
oracle.mft.adapter.ftp	Not Applicable	INFO	SEVERE
oracle.mft.adapter.ftp.inbound	Not Applicable	INFO	SEVERE
oracle.mft.adapter.ftp.outbound	Not Applicable	INFO	SEVERE
oracle.mft.adapter.ftp.connection	Not Applicable	INFO	SEVERE
oracle.mft.adapter.jms	Not Applicable	INFO	SEVERE
oracle.mft.adapter.jms.inbound	Not Applicable	INFO	SEVERE
oracle.mft.adapter.jms.outbound	Not Applicable	INFO	SEVERE
oracle.mft.adapter.jms.connection	Not Applicable	INFO	SEVERE
oracle.mft.adapter.jms.transaction	Not Applicable	INFO	SEVERE

Viewing Oracle Managed File Transfer Log Messages

Use the Oracle Enterprise Manager Fusion Middleware Control console to view messages in the Oracle Managed File Transfer log files.

The steps for this process are:

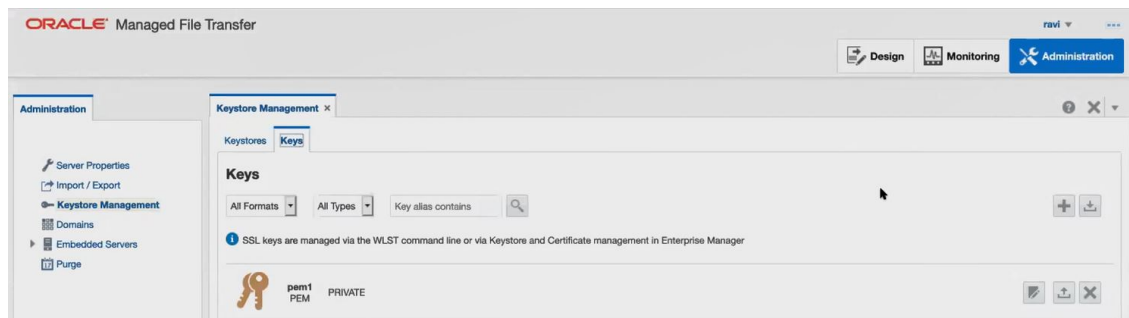
1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the WebLogic Domain node.
3. Expand the node for the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.

For example, the domain might be `soainfra` or `base_domain`.

4. Right-click the MFT server and select **Logs**, then **View Log Messages**.
The Log Messages page appears.
5. To display a list of MFT log files, click **Target Log Files**.
The Log Files page appears.
6. To View a log file, select the file and click **View Log File**.
The View Log File page appears.
7. To view specific types of messages, specify search criteria and click **Search**.

Managing Keystores Using the Oracle Managed File Transfer Console

Keys and certificates are used to digitally sign and verify data and achieve authentication, integrity, and privacy in network communications. Use the **Keystore Management** tab on the Administration page to manage security keys using the Oracle Managed File Transfer console.



The **Keystore Management** tab presents the following subtabs:

Keystores Tab

The **Keystores** tab lists the Default Keystore, PGP Keystore, and SSH Keystore. Access to a keystore requires a password. See [Using the Keystores Tab](#).

Keys Tab

The **Keys** tab lists the existing keys in the system. Use the **Keys** tab to manage SSH, PEM, and PGP keys. See [Using the Keys Tab](#).

Using the Keystores Tab

The **Keystores** tab lists the **Default Keystore**, **SSH Keystore**, and **PGP Keystore**. Access to a keystore requires a password which is defined at the time the keystore is created, by the person who creates the keystore, and which can only be changed by providing the current password.

In addition, each private key in a keystore can be secured by its own password.

Default (SSL) Keystore

Oracle MFT uses SSL certificates for embedded server security to start an embedded FTPS server. The default feystore stores the SSL certificates. An SSL certificate is used when there

is an inbound transaction request when connecting to an embedded FTPS server. A private key and digital certificate provide identity for the server. The SSL certificates have private and public keys.

When logging in to an external FTPS, you must import the SSL certificate of the remote FTPS server to the SSL keystore to connect.

For steps to create the SSL keystore and keys using WLST commands and the Oracle Managed File Transfer console, see [Configuring the SSL Keystore](#).

SSH Keystore

Oracle MFT uses SSH keys for embedded SFTP server security. The SSH keystore is used to store private and public keys.

For an embedded SFTP server, private SSH keys are used to start the embedded SFTP server and public keys are used to enable key-based authentication to the embedded SFTP server. To enable key-based authentication for an embedded SFTP server, you must import the public key into the SSH keystore and specify the private key at the SSH client. While importing, the public key alias should be same as the user name.

For a remote SFTP server, private keys are used to authenticate to log in to a remote server. The private key of the SSH key pair should be imported to the SSH keystore with any alias and that alias must be selected in the SFTP source/target and the public key will be shared with the remote SFTP server.

For steps to create the SSH keystore and keys using WLST commands and the Oracle Managed File Transfer console, see [Configuring the SSH Keystore](#).

PGP Keystore

The PGP keystore is used for encryption and decryption of the source and targets. The private keys are used to decrypt the payload and the public keys are used to encrypt the payload.

With the public key encryption, a public and a private key are generated for a server. Data encrypted with the public key can only be decrypted using the corresponding private key and vice versa.

For steps to create the PGP keystore and keys using WLST commands and the Oracle Managed File Transfer console, see [Configuring the PGP Keystore](#).

Using the Keys Tab

Use the **Keys** tab to create, import, export, delete, and update keys. The **Keys** tab lists the existing keys in the system.

You can filter the list of existing keys using the drop-down lists:

- **Format:** select **SSH**, **PEM**, **PGP**, or **All Formats**.
- **Type:** select **Private**, **Public**, or **All Types**. For an RSA key of PEM format, only **Private** is valid.



Refer to the following topics to manage keys on the **Keys** tab:

- [Creating a Key](#)
- [Exporting a Key](#)
- [Importing a Key](#)
- [Updating a Key](#)
- [Deleting a Key](#)

Creating a Key

You can create a new key using the **Keys** tab on the **Keystores Management** page.

Notes:

- To generate a private RSA key of PEM format, which is used to connect to Oracle Cloud Infrastructure when the OCI Storage Cloud Service type is selected as a source or target, you cannot use the Oracle Managed File Transfer console or the WSLT `generateKeys` command. Instead, you can use an external key generation application, such as `ssh-keygen`, or follow the steps in [How to Generate an API Signing Key](#) in the Oracle Cloud Infrastructure documentation. Then, you can import the RSA key of PEM format.
- You can also use the Oracle WebLogic Scripting Tool (WLST) to create a key. See `generateKeys` in [MFT WLST Command Summary](#).

To create a key:

1. From the Administration page, select **Keystore Management**, then select the **Keys** tab.
2. Click the **Create Key**



icon on the right side of the page to create a key.

3. In the Create Key dialog, enter the following details:
 - **Alias:** alias name.
 - **Format:** select **PGP** or **SSH**.
 - **Key size:** enter a valid integer between 1024 and 8192.
 - **Password:** optionally, enter a password.
 - **Confirm Password:** if entered, confirm the password.
 - **Identity:** if creating PGP key, enter a default identity, such as an email ID.
 - **Import Private Key:** select if importing a private key.
4. Click **Create**.

A Download Key dialog is displayed with a link to download the generated key.
5. Click the **Download** link to download the zip file containing the generated key.

Exporting a Key

You can export a key using the **Keys** tab on the **Keystores Management** page.



Note:

You can also use the Oracle WebLogic Scripting Tool (WLST) to export a key. See `exportCSFKey` in [MFT WLST Command Summary](#).

To export a key:

1. From the Administration page, select **Keystore Management**, then select the **Keys** tab.
2. Click the **Export Key**



icon to the right of the key you want to export.

Depending on your system settings, a window opens prompting you for the location to save the key file or the file is automatically downloaded to your Downloads directory.

Importing a Key

You can import a new key using the **Keys** tab on the **Keystores Management** page.



Notes:

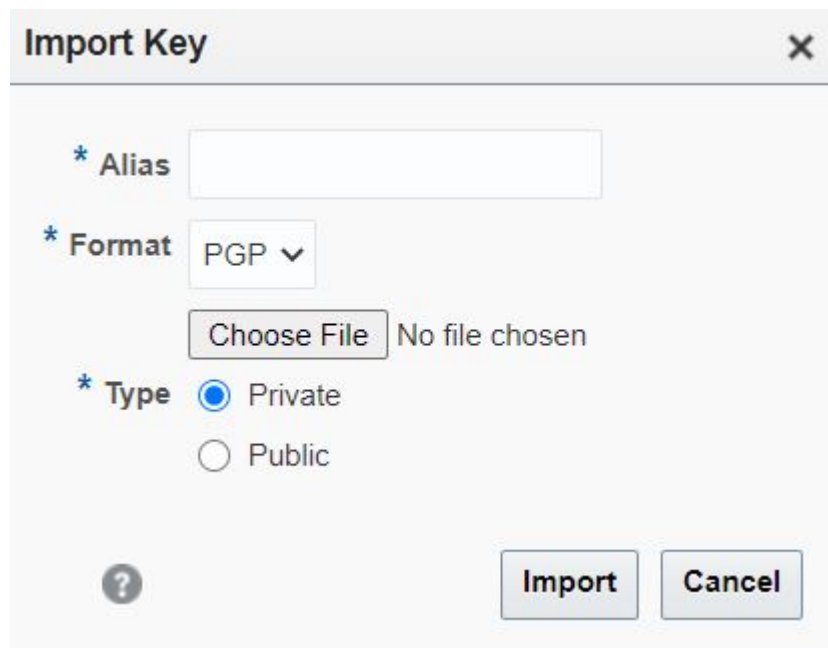
- Before you can use the OCI Storage Cloud Service type as a source or target, you must import a private RSA key of PEM format to connect to Oracle Cloud Infrastructure.
- You can also use the Oracle WebLogic Scripting Tool (WLST) to import a key. See `importCSFKey` in [MFT WLST Command Summary](#).

To import a key:

1. From the Administration page, select **Keystore Management**, then select the **Keys** tab.
2. Click the **Import Key**



icon on the right side of the page.



3. In the Import Key dialog, enter the following details:
 - **Alias:** alias name.
 - **Format:** select **SSH**, **PEM**, or **PGP**.

 **Note:**

An RSA key of PEM format is used to connect to Oracle Cloud Infrastructure when the OCI Storage Cloud Service type is selected as a source or target.

- **Choose File:** browse to select the key file.
 - **Type:** select **Private** or **Public**.
4. Click **Import** to import the key.

Updating a Key

You can update an existing key using the **Keys** tab on the **Keystores Management** page.

 **Note:**

You can also use the Oracle WebLogic Scripting Tool (WLST) to update a key. See `updateCSFKey` in [MFT WLST Command Summary](#).

To update an existing key:

1. From the Administration page, select **Keystore Management**, then select the **Keys** tab.

2. Click the **Update Key**



icon to the right of the key you want to update.

3. In the Update Key dialog, enter the following details:
 - **Alias:** alias name.
 - **Choose File:** browse to select the updated key file.
4. Click **Update** to update the key.

Deleting a Key

You can delete an existing key using the **Keys** tab on the **Keystores Management** page.



Note:

You can also use the Oracle WebLogic Scripting Tool (WLST) to delete a key. See `deleteCSFKey` in [MFT WLST Command Summary](#).

To delete a key:

1. From the Administration page, select **Keystore Management**, then select the **Keys** tab.
2. Click the **Delete Key**



icon to the right of the key you want to delete.

3. In the Confirm dialog, click **Yes**.

Incrementally Moving Oracle MFT Metadata with Configuration Plans

You can substitute key protocol attributes when you move MFT metadata from development or test environments to the production environment.

As you move metadata from one environment to another (for example, from testing to production), you may need to modify environment-specific values. Configuration plans enable you to modify these values using a single text (XML) file. The configuration plan can be created by using the `export wlst` command or by exporting it from the UI. The default generated configuration plan XML contains the current values. When you are ready to move the file to production you can modify the config plan xml to provide the substitute string for values that you want in the production environment. When calling the `import` command in the production system, you can optionally specify the modified config xml file to specify the string substitutions. To verify the modified configuration plan the `previewMode` option can be used. This option is available only in the `wlst` command not in the UI.

A new parameter, `configFile` has been added to the existing `import wlst` commands, which gives you the option to specify the string substitutions via a `configPlan` xml for key attributes.

Protocol	Attributes
File	folder
FTP	host, controlPort, user, password, folder
SSH-FTP	host, port, user, password, folder, authenticationType, privateKeyFile, publicKeyFile
Embedded-FTP	folder
Embedded-SSH-FTP	folder
WS(SOA,SOAP,ODI,OSB)	url, service, port, action
B2B	tpChannelName

Understanding When to Use Configuration Plans and T2P Migration

When you need to migrate data in MFT, you have two options, T2P migration or MFT Configuration Plans.

T2P is for one-time migration of whole deployments utilizing an MFT Plugin to the T2P Framework. The movement scripts are intended for moving to a new target environment. They do not support moving artifacts to an already existing environment nor do they allow moving artifacts incrementally.

For more information about the T2P migration scripts, see *Moving from a Test to a Production Environment in Administering Oracle Fusion Middleware*.

The MFT Config Plan is for incremental migration including artifact level properties (Sources, Targets, Transfers). You utilize the Config Plan in cases such as:

- You need to migrate *transfer1* from test to production and change the hostname from testHost1.com to partner1.com.
- The password is different from one environment to another so the MFT Config Plan provides the option to change passwords during import
- The PGP key alias is different from one environment to another so the MFT Config Plan provides the option to change key alias during import

Global Substitution for Attributes

You can globally substitute any of the attribute values apart from name or any internal field. The following are two examples of global string replacements:

```
<searchReplace>
```

```
    <search>http://my-dev-server</search>
```

```
    <replace>http://my-test-server</replace>
```

```
</searchReplace>
```

```
<Source name= "*">
```

```
<searchReplace>
```

```
    <search>http://my-dev-server1</search>
```

```
<replace>http://my-test-server1</replace>

<searchReplace>

<Source>
```

Changing Key Attributes for a Given Source or Target

The ConfigPlan xml lists the key attributes for all the sources and targets and it provides the option to change the value for a specific artifact. For example:

```
<Source name="PO Source">

<binding type="ftp">

  <attribute name="port">

    <replace>8888</replace>

  </attribute>

  <attribute name="host">

    <replace>example.com</replace>

  </attribute>

</binding>

</Source>
```

In case of a conflict, the order of preference is:

1. Artifact specific change
2. Artifact specific global rule
3. Global replacement

Basic validation is done for all substitution fields. Complete validation is done only during deployment. Basic validation includes:

- Type validation : example only true/false will be allowed for Boolean fields
- Restrictions check : if the field is bounded by certain specified values then only those values will be allowed

Specifying the Artifacts to Include in the Import

The config plan xml lets you specify the artifacts to include in the import. For example:

```
<transfers>

  <include>

    <transfer name="PO transfer"/>

  </include>
```

<transfers>

Password Change

A new wlst command `createMftCredential` returns and/or prints the CSF reference to the generated key. Once the key is generated in the **target environment** the same reference can be used in the config plan to replace the password fields.



Note:

Clear passwords are not supported.

Preview Imports

The preview mode option for the import command does not import metadata; it simply list the artifacts that will be exported and all attributes which will be overridden from the export zip because of the config plan. The default value is false.

Modified WLST Commands

Command	Details	Syntax and Parameters	Examples
<code>exportMetadata</code>	Export (only TIP version) all the MFT metadata to given file	<pre>exportMetadata ('<ArchiveFile>', generateConfigPlan, longFormat)</pre> <p>ArchiveFile: File location for exporting metadata</p> <p>generateConfigPlan(optional, default false): Indicates whether to generate the mftConfig XML. Config plan is generated in the same folder where archive file is generated</p> <p>longFormat(optional, default false): If <i>true</i>, most of the attributes are included in the config plan xml; otherwise, only the key attributes are listed in the config plan XML</p>	<pre>wls:/mydomain/ serverConfig> exportMetadata("/tmp/ mft/mds.zip")</pre> <pre>wls:/mydomain/ serverConfig> exportMetadata("/tmp/ mft/mds.zip", true)</pre> <pre>wls:/mydomain/ serverConfig> exportMetadata("/tmp/ mft/mds.zip", true, true)</pre>

Command	Details	Syntax and Parameters	Examples
exportTransferMetadata	Export (only TIP version) the given transfer and related metadata to a given file.	<p>exportTransferMetadata(' <ArchiveFile>', '<TransferName>', generateConfigPlan, longFormat)</p> <p>ArchiveFile: File location for exporting metadata</p> <p>TransferName: Name of transfer to be exported</p> <p>generateConfigPlan(optional, default false): Indicates whether to generate the mftConfig XML. Config plan is generated in the same folder where archive file is generated</p> <p>longFormat(optional, default false): If <i>true</i>, most of the attributes are included in the config plan xml; otherwise, only the key attributes are listed in the config plan XML</p>	<pre>wls:/mydomain/ serverConfig> exportTransferMetadata('/tmp/mft/mds-xfer.zip', 'cotsmart-xfer') wls:/mydomain/ serverConfig> exportTransferMetadata('/tmp/mft/mds-xfer.zip', 'cotsmart-xfer', true) wls:/mydomain/ serverConfig> exportTransferMetadata('/tmp/mft/mds-xfer.zip', 'cotsmart-xfer', true, true)</pre>

Command	Details	Syntax and Parameters	Examples
exportDeployedArtifact	Export a deployed artifact from a given mds label.	<p>exportDeployedArtifact(<ArtifactType>,'<ArtifactName>','<Label>','<ArchiveFilePath>',generateConfigPlan,longFormat)</p> <p>ArtifactType - SOURCE / TRANSFER / TARGET ArtifactName: Name of the artifact to exported Label: MDS label to be exported ArchiveFilePath: File to which artifact is to be exported. It should be a zip file and should not exist. generateConfigPlan(optional, default false): Indicates whether to generate the mftConfig XML. Config plan is generated in the same folder where archive file is generated longFormat(optional, default false): If true, most of the attributes are included in the config plan xml; otherwise, only the key attributes are listed in the config plan XML</p>	<pre>wls:/mydomain/ serverConfig> exportDeployedArtifact('TARGET','cotsmart- file- tgt','soa_mft-2012-12 -07 22:24:09.383','/tmp/ export/cotsmart- file.zip') wls:/mydomain/ serverConfig> exportDeployedArtifact('TARGET','cotsmart- file- tgt','soa_mft-2012-12 -07 22:24:09.383','/tmp/ export/cotsmart- file.zip', true) wls:/mydomain/ serverConfig> exportDeployedArtifact('TARGET','cotsmart- file- tgt','soa_mft-2012-12 -07 22:24:09.383','/tmp/ export/cotsmart- file.zip', true,true)n</pre>

Command	Details	Syntax and Parameters	Examples
importMetadata	Import metadata from a given file location. Optionally mft config plan can be specified to change the parameter during import. Command can be run in preview mode as well.	<pre>importMetadata ('<ArchiveFile>', <MftConfigPlanXML>, previewMode)</pre> <p>ArchiveFile: File location for importing metadata configFileLocation(optional): File location for mftConfig XML previewMode(optional, default false): If <i>true</i>, no metadata will be imported. It can be used to determine which attributes will be changed because of config plan.</p>	<pre>wls:/mydomain/ serverConfig> importMetadata ("tmp/ mft/mds.zip") wls:/mydomain/ serverConfig> importMetadata ("tmp/ mft/ mds.zip", "tmp/mft/ mftconfigplan.xml")</pre> <p>This will import the metadata and override the values using the config plan XML.</p> <pre>wls:/mydomain/ serverConfig> importMetadata ("tmp/ mft/ mds.zip", "tmp/mft/ mftconfigplan.xml", false)</pre> <p>It will not import the metadata. It provides info on which attributes will be changed using of config plan.</p>
createMftCredential	Create the credential for mftapp.	<pre>createMftCredential (password, key)</pre> <p>password: password for which credential needs to be created for mftapp. key(optional) : key for the credential</p>	<pre>wls:/mydomain/ serverConfig> createMftCredential (' mypass')</pre> <pre>wls:/mydomain/ serverConfig> createMftCredential (' mypass', 'mykey')</pre>

Related Topics

- [WLST Commands](#)
Use WLST (Oracle WebLogic Scripting Tool) commands to perform various Oracle Managed File Transfer operations.

Adding a Purge Schedule

Use the Purge option from the Administration page to purge old instance and payload data from Oracle Managed File Transfer, thereby improving performance. The purge schedule is in addition to the seeded default purge.

You can purge data instantly or create a schedule to purge data. You can specify when to start and stop the purge, the frequency of the schedule purge and provide additional criteria to filter

the instances to be purged. The criteria may include transfer status, transfer names, and the time range of the instances to be purged.

**Note:**

The Purge can be executed only by an admin user/administrator.

For information on creating a purge schedule, see [Creating a New Purge Schedule](#).

For information on creating a Run Now purge, see [Creating a Run Now Purge](#).

For information on modifying a purge schedule, see [Modifying or Deleting an Existing Purge Schedule](#).

You can create or modify a purge schedule by using WLST commands. For more information, see [MFT WLST Commands Summary](#).

Creating a New Purge Schedule

The Purge tab can be accessed from the Administration page. You can view the list of existing purge schedules. You can **Create**, **Delete**, **Edit** or **Activate/Deactivate** the purge schedule.

You can create an automated purge schedule to run at a predefined time and frequency. When purge schedule is executed at the scheduled date and time, the system will purge data that meets the filter criteria of status, transfer names and retention period.

**Note:**

The default Java Transaction API (JTA) time-out is 30 seconds, which is set in the Weblogic Admin console. Sometimes the purge operation may take more time to get executed based on data size and you may get a transaction time-out exception. To avoid exception errors, use the MBean property `purgeTransactionTimeout` to modify the default JTA time-out for purge operation. See [MFT WLST Command Summary](#) for more information.

To add a new schedule purge:

1. From the Administration page, select the **Purge** tab.

You can view the Default Schedule and purge schedules created.

2. Select the **Add '+'** icon to create a new purge schedule.

The Add Purge Schedule dialog opens.

3. Specify the following details in the Add Purge schedule dialog:

- **Name** - Specify a name for the schedule purge.
- **Schedule Start Date** - The schedule start date.
- **Schedule End date** (Optional) - The schedule end date on which the schedule stops. If the end date is not specified, the schedule will purge data continuously at the scheduled date and time.
- **Schedule Time** - Specify the time when the purge starts. By default, it is set to 12 am.

- **Frequency** - Specify the frequency of the schedule. Options are daily, weekly, monthly or yearly.
 - **Retention Period** - Specify the retention period for the old data in days. It specifies the time period of the instances that should not be included in the purge, it is calculated from the schedule run date. Default period is 7 days.
 - **Status** - Status of the instances that will be purged. Options - Completed and/or Failed. One of the options must be checked.
 - **Transfer Filter** - Name of transfer whose instance needs to be purged. If the Transfer name is not specified, the action would be applied to all instances for all transfers. When clicked, **Select Transfer** dialog appears where you can select the Transfer Names.
 - **Comments** - Provide comments.
4. Select **OK** to continue or **Cancel** to cancel the action.
A confirmation dialog opens.
 5. Select **OK** on the confirmation dialog.
The newly created purge schedule is displayed in the list of Purge schedules with **Next Execution Time** and **Activation Status** as Active.

Creating a Run Now Purge

You can purge instance data immediately by selecting the **Run Now** option.

To create a Run Now Purge:

1. From the Administration page, select the **Purge** tab.
2. Select the **Run Now** tab.
The Run Now dialog opens.
3. Specify the following details in the Run Now dialog:
 - **Retention Period** - Specify the retention period of the old data in days. It specifies the time period of the instances that should be purged calculated from the schedule run date. Default period is 7 days.
 - **Status** - Status of the instances that will be purged. Options - Completed and/or Failed. One of the options must be checked.
 - **Transfer Filter** - Name of transfer whose instance needs to be purged. If the Transfer name is not specified, the action would be applied on all instances for all transfers. When clicked, **Select Transfer** dialog appears where you can select the Transfer Names. Select **OK** to continue.
 - **Comments** - (Optional) provide comment.
 - **Execute** - Select to execute the Purge.

On execution, the instance data that meets the filter criteria is deleted immediately.

Modifying/Deleting an Existing Purge Schedule

You can modify or delete an existing purge schedule. To modify an active purge schedule, change the status to inactive and then modify the purge. However, if you edit an active schedule, a confirmation dialog opens up asking if you want to deactivate and edit the active schedule, click **OK** to continue.

You cannot edit the name of the purge schedule, all other details such as date, time, frequency, or retention period can be edited.



Note:

You cannot delete the Default Purge schedule or an active purge schedule.

If the selected purge schedule is in inactive status, you can delete, edit, or change the status. If the purge schedule is in active status, you can edit it only after changing the status to inactive.

When deactivating an active schedule, a confirmation message confirms if you want to go ahead with deactivation. Select OK to deactivate. This will disable the purge and change it to inactive status.

When you delete a schedule, it is removed from the list of purge schedules.

7

Administering Oracle Managed File Transfer Embedded Servers

Learn how to administer the FTP and sFTP (SSH-FTP) servers embedded by Oracle Managed File Transfer.

This chapter includes the following sections:

- [About Embedded FTP and sFTP Servers](#)
- [Embedded Server Configuration](#)
- [Starting and Stopping Embedded Servers](#)
- [Managing Embedded Servers and High Availability](#)
- [Supported FTP and sFTP Commands](#)

For information about administering the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer, see [Administering Oracle Managed File Transfer](#).

About Embedded FTP and sFTP Servers

Oracle Managed File Transfer includes built-in FTP and sFTP servers, which handle many of the types of file transfers performed. Uploading a file into one of the embedded server directories is necessary to transfer a file using one of the embedded servers.

These embedded servers have their own file system directories for sending and receiving files. The default root directory location for both the FTP and sFTP servers is `WLS_Home/user_projects/domains/base_domain/mft/ftp_root`. To change this location, see [Other Embedded Server Settings](#).

The `payloads` directory under the root is for files accessed by external systems such as SOA.

The FTP embedded server is enabled by default, but its security features are disabled. The sFTP server is disabled by default.

Oracle Managed File Transfer supports the use of any compliant FTP or sFTP client.

Note:

Files present in an embedded FTP or sFTP source directory before the source is deployed or enabled are ignored. Only files uploaded to the directory after deployment or enabling are picked and transferred.

Security

Oracle Managed File Transfer embedded servers can restrict user access to the server file systems. For more information, see [Embedded Server Security](#) and [Embedded Server User Access](#).

Archiving and Purging Transfers and Files

To clear space in the embedded server file systems, you can use WLST commands to archive and purge transfer instances and their associated files. For more information, see MFT Archive and Restore Commands and MFT Purge Commands in *WLST Command Reference for SOA Suite*.

Single Source Instance can be purged from the user interface using the Purge option in the Source Instance of the Monitoring page. For more information, see [Interpreting Artifact Instance Messages](#).

Files uploaded to embedded server directories for which no sources are configured cannot be purged or archived. Only files associated with a transfer instance can be purged or archived.

Embedded Server Configuration

You can set the Embedded server configuration settings in the Embedded Servers, Embedded Server Ports, and Embedded Server User Access tabs from the Administration page.

For information about these settings, see the following sections:

- For Embedded Server settings, see [Embedded Server Security](#) and [Other Embedded Server Settings](#).
- For Embedded Server Ports settings, see [Re-configuring the Port](#).
- For Embedded Server User Access settings, see [Embedded Server User Access](#).

Re-configuring the Port

You can reconfigure the ports of the Oracle Managed File Transfer embedded servers.

Embedded servers are not the same thing as Oracle WebLogic Server managed servers. The MFT embedded FTP and sFTP servers are services that run on the Oracle WebLogic Server managed server(s) dedicated to Oracle Managed File Transfer.

The steps for this process are:

1. On the left pane of the Administration page, click the arrow to the left of **Embedded Servers**.
The **Ports** and **User Access** items appear.
2. Click **Ports**.
The Embedded Server Ports tab opens.
3. Change the **Configured Port** value for each **Server Instance** and **Service** (FTP or sFTP) combination you want to reconfigure.
4. Click **Save**.
5. Click **Restart**, **Start**, or **Stop**.
6. Click the **Refresh** icon to update the table. Verify that the new **Server Status** for all servers is RUNNING and that the new **Running Port** values are the values you configured.

Path Separators for Remote FTP and sFTP Servers

If the path separator on a remote FTP server is not the conventional forward slash (/), you must specify the **FTP Path Separator** when you configure FTP Remote or sFTP Remote sources and targets.

For more information about other settings you can edit after you create an FTP Remote or sFTP Remote source or target, see the *Oracle Fusion Middleware MFT Composer Help Online Help*.

Other Embedded Server Settings

[Table 7-1](#) lists the FTP embedded server settings not related to security. These settings are on the Administration page, Embedded Servers tab, and FTP subtab.

Table 7-1 FTP Embedded Server Settings

Setting	Description
Root Directory	<p>Specifies the root directory of the FTP server. The default is <code>WLS_Home/user_projects/domains/base_domain/mft/ftp_root</code>. You must set it to a shared location if multiple Oracle WebLogic Server instances run in a cluster.</p> <p>To change the root directory:</p> <ol style="list-style-type: none"> 1. Stop the Oracle WebLogic Server managed server(s) dedicated to Oracle Managed File Transfer. 2. Change the Root Directory setting. 3. Move the directories and files under the root directory. 4. Restart the managed server(s). <p>See Re-configuring the Port for how to stop and start the managed server(s).</p>
Enabled	Enables the sFTP server if checked. The default is enabled (checked).
Maximum Logins	Specifies the maximum number of concurrent users. The default is 10.
Maximum Login Failures	Specifies the maximum number of login failures after which a connection is closed. The default is 3.
Maximum Concurrent Requests	Specifies the maximum number of concurrent requests the sFTP server can accept. The default is 10.
Idle Timeout	Specifies the time in seconds that the server can be idle before the connection ends and the user must log in again. The default is 600, equivalent to 10 minutes.
Active Connection: Port Range Start, Port Range End	In active mode, the client establishes the command channel. The server establishes the data channel between a server port in the range from Port Range Start to Port Range End and a client port that the client specifies using the <code>PORT</code> command.
Enable	Enables the FTP server if checked. The MFT server does not start a disabled FTP server during initialization. The default is disabled (unchecked).
IP Check	Finds out whether the IP address for the data connection is the same as for the control socket if checked. FTP uses two channels between client and server, which are separate TCP connections. The command channel is for commands and responses. The data channel is for transferring files. The default is disabled (unchecked).

Table 7-1 (Cont.) FTP Embedded Server Settings

Setting	Description
Passive Connection:	In passive mode, the client establishes both the command and data channels.
Port Range Start, Port Range End	The server tells the client which port in the range from Port Range Start to Port Range End to use for the data channel.

[Table 7-2](#) lists the sFTP embedded server settings not related to security. These settings are on the Administration page, Embedded Servers tab, and sFTP subtab.

Table 7-2 sFTP Embedded Server Settings

Setting	Description
Root Directory	Specifies the root directory of the sFTP server. The default is <i>WLS_Home/user_projects/domains/base_domain/mft/ftp_root</i> . You must set it to a shared location if multiple Oracle WebLogic Server instances run in a cluster. To change the root directory: <ol style="list-style-type: none"> 1. Stop the Oracle WebLogic Server managed server(s) dedicated to Oracle Managed File Transfer. 2. Change the Root Directory setting. 3. Move the directories and files under the root directory. 4. Restart the managed server(s). See Re-configuring the Port for how to stop and start the managed server(s).
Enabled	Enables the sFTP server if checked. The default is enabled (checked).
Maximum Concurrent Requests	Specifies the maximum number of concurrent requests the sFTP server can accept. The default is 10.
Maximum Login Failures	Specifies the maximum number of login failures after which a connection is closed. The default is 3.
Idle Timeout	Specifies the time in seconds that the server can be idle before the connection ends and the user must log in again. The default is 600, equivalent to 10 minutes.

For information about the embedded server settings related to security, see [Embedded Server Security](#).

Starting and Stopping Embedded Servers

You can start or stop the Oracle Managed File Transfer embedded servers from the Administration page.

The steps for this process are:

1. Open the Embedded Servers tab on the Administration page.
2. Select the FTP or sFTP subtab.
3. Look at the **Start** and **Stop** buttons. If **Start** is grayed out, the server is running and you can **Stop** or **Restart** it. If **Stop** is grayed out, the server is not running and you can **Start** it.
4. Click **Start**, **Stop**, or **Restart**.

To restart all embedded servers, click **Restart All**.

To start, restart, or stop the Oracle WebLogic Server managed server(s) dedicated to Oracle Managed File Transfer, see [Oracle WebLogic Server Startup and Shutdown](#).

Managing Embedded Servers and High Availability

When you set up high availability for Oracle Managed File Transfer, the embedded servers participate in the load balancing, failover, and other features.

For more information, see [Managing Multiple Weblogic Servers and High Availability](#).

Supported FTP and sFTP Commands

Oracle Managed File Transfer supports FTP and sFTP commands.

Supported FTP commands are:

- **APPE**: Appends data to the end of a file on the remote host.
- **AUTH**: Establishes SSL encrypted session. Only the SSL type is supported.
- **CDUP**: Changes to the parent directory.
- **CWD**: Changes the working directory. If the directory name is not specified, the root directory (/) is assumed.
- **LIST**: Lists files. Must be preceded by a **PORT** or **PASV** command.
- **MKD**: Creates a directory.
- **MLSD**: Similar to **LIST**.
- **NOOP**: No operation.
- **PASS**: Specifies the password. Must be immediately preceded by **USER**.
- **PASV**: Listens on a data port.
- **PORT**: Specifies the data port.
- **PROT**: Specifies the data channel protection level.
- **PWD**: Returns the name of the current working directory.
- **REST**: Specifies the marker position from which the file transfer is to be restarted.
- **RETR**: Transfers a copy of the file specified as the argument.
- **SIZE**: Returns the size of the file in bytes.
- **STOR**: Accepts and stores the data received as a file.
- **TYPE**: Specifies the file type, `ascii` (the default) or `binary`.
- **USER**: Specifies the user name.
- **QUIT**: Logs out.
- **DELE**: Deletes the file specified by the provided path.
- **RMD**: Removes a directory.
- **RNFR**: Renames from the specified name.
- **RNTO**: Renames to the specified name.

- **STOU:** Stores a unique file to this directory.

Supported sFTP commands are:

- **cd:** Changes the remote working directory.
- **get:** Downloads a file from a remote directory to a local directory.
- **ls:** Lists the contents of a remote directory.
- **mkdir:** Creates a remote directory.
- **mv:** Moves or renames a remote file.
- **put:** Uploads a file from a local directory to a remote directory.
- **pwd:** Prints a remote working directory.
- **rm:** Removes a remote file.
- **rmdir:** Removes a remote directory.
- **quit, bye:** Ends and disconnects a client or user session.

Oracle Managed File Transfer does not support shell command execution.

8

Oracle Managed File Transfer Security

Learn how to keep Oracle Managed File Transfer and its embedded FTP and SFTP servers secure.

This chapter includes the following sections:

- [User Authentication and Authorization](#)
- [Embedded Server Security](#)
- [Remote SFTP Server Security](#)
- [Integrating with Oracle Access Manager 11g for Single Sign-On](#)
- [Message Encryption Using PGP](#)
- [FIPS 140 Compliance](#)
- [Creating an Oracle Managed File Transfer Stripe](#)
- [Managing Keystores Using WLST Commands](#)
- [Enabling Security Audit Logging](#)
- [OWSM Security Policy Attachment](#)
- [Configuring SSL only Domain for Oracle Managed File Transfer](#)

User Authentication and Authorization

You can configure users, grant them access to Oracle Managed File Transfer and give permissions to embedded FTP and SFTP server directories and transfer payloads.

Note:

If a user's permissions have changed when the user is already logged in, the changes will be effective only on the next login. Once a user is authenticated, the permissions of the user will not change until the user logs out and logs in again. This is applicable for all the console, WLST, RESTful, and embedded server operations such as deploy, enable/disable, import/export or read/write embedded server operations.

Note:

The group or role association for a user or group from WebLogic console or Enterprise Manager are not immediately reflected in the managed server. There is a delay of approximately 10 minutes to reflect the role membership in the MFT application. So any new group association will be delayed to take effect while executing any MFT security operation.

Configuring Users

You configure Oracle Managed File Transfer users and assign them to groups in the Oracle WebLogic Server Administration Console.

 **Note:**

Oracle Managed File Transfer interacts with Oracle Enterprise Scheduler Service through the `OracleSystemUser`. Do not delete this user. If you do, clicking **add schedule** in a transfer configuration will result in an `OracleSystemUser does not exist` message, and **Schedule Details** may be blank in monitoring reports. For more information about transfer schedules, see [Setting Up Schedules](#).

The steps for this process are:

1. Access the Oracle WebLogic Server console using a URL that includes the Oracle WebLogic Server hostname and the console port:

```
http://wls-hostname:console-port/console
```

For example:

```
http://localhost:7011/console
```
2. Log in using the Oracle WebLogic Server admin username and password.
3. In the left pane of the Oracle WebLogic Server Administration Console, select **Security Realms**.
4. On the Summary of Security Realms page, select the name of the realm (for example, **myrealm**).
5. On the Settings for *Realm Name* page, select **Users and Groups > Users**.
6. Click **New**.
7. In the **Name** field of the Create New User page, enter a unique alphanumeric name for the user.
8. In the Description field, enter a description. The description might be the user's full name. This is optional.
9. In the **Provider** drop-down list, select **DefaultAuthenticator**.
10. In the **Password** field, enter a password for the user.

The minimum password length is 8 characters. Do not use the username/password combination `weblogic/welcome1` in production.
11. Re-enter the password for the user in the **Confirm Password** field.
12. Click **OK** to save your changes.
13. Click the name of the new user in the User table.
14. On the Settings for *User Name* page, select **Groups**.
15. Select a group or groups from the **Available** list box and move them to the **Chosen** list box. A user can be a member of more than one group.

See [Table 8-1](#) for MFT console access groups and [Table 8-2](#) for MFT embedded server access groups.

16. Click **Save**.

For complete details, see *Create Users and Add Users to Groups* in the *Oracle WebLogic Server Administration Console Online Help*.

Oracle Managed File Transfer Console Access

Users log in to the Oracle Managed File Transfer console using the name and password assigned to them through the process described in [Configuring Users](#). The Oracle Managed File Transfer page on which the user starts depends on the user's group:

- Administrators and monitors start on the Monitoring page.
- Deployers start on the Designer page.

A user assigned to both the Deployers and Monitors groups starts on the Designer page.

[Table 8-1](#) lists the roles, groups, and permitted actions for user access to the MFT console. Console access is based only on roles. Embedded server access roles and groups do not determine console access.

Table 8-1 MFT Console Roles, Groups, and Permissions

Role	Groups with Role	Console Actions Permitted
MFTAdmin	Administrators, OracleSystemGroup	Import, Export, Purge, Design, Deploy, Monitor, Resubmit, Pause-Resume, Retry, Disable, Enable, StartES, StopES (all actions)
MFTMonitor	Monitors	Monitor, Resubmit, Pause-Resume, Retry, Disable, Enable, StartES, StopES
MFTDesigner	Deployers	Design, Deploy

Embedded Server User Access

You can grant users access to embedded FTP and sFTP server directories.

To use WLST to configure embedded server user access, see [Using WLST Commands with Oracle Managed File Transfer](#) and MFT Embedded Server Commands in *WLST Command Reference for SOA Suite*.

To grant access to embedded FTP and sFTP server directories:

1. On the left pane of the Administration page, click the arrow to the left of **Embedded Server**.

The **Ports** and **User Access** items appear.

2. Click **User Access**.

The Embedded Server User Access tab opens.

3. Select **User**, **Group**, or **Role**, then type a user, group, or role name in the text field. You must type at least three letters. Any matches are displayed below the text field. Click the match you want to add.

You configure users, groups, and roles using the process described in [Configuring Users](#).

 **Note:**

If **Enable screen reader mode** is selected in Accessibility Preferences, you must type the full user, group, or role name. See [Setting Language, Time Zone, and Accessibility Preferences](#) for more information.

4. Click the **Add Folder (+)** icon.
The user's default folder is added to the table with **Set As Home Folder** selected.
5. To add other folders, click the **Search** icon.
 - a. Type a directory under which to search in the **Available Folders** text box.
 - b. Click the arrow to the right of the **Available Folders** text box.
Subdirectories of the search directory are displayed.
 - c. Click the arrow to the left of a directory to display further subdirectories.
 - d. To select a directory, check its box.
 - e. Click **Add Selected**.
6. Select **Set As Home Folder** to assign a different home folder. This is optional.
If **Set as Home Folder** is selected, the user is placed in the Home Folder when they log on to the embedded server. If the home folder does not exist, it is created at login.
7. Set the following permissions for each row. To set a permission for all rows, check or uncheck the box in the column header.
 - Access Subfolders: Applies the same permission settings to all subfolders.
 - Read: Allows viewing of file contents.
 - Write: Allows modification of file contents.
 - Delete: Allows file deletion.
 - List: Allows viewing of directory contents.
8. Click **Save**.

To undo all permission changes for a specific user since the last Save, click **Reset All**. To undo all changes for all users since the last Save, click **Revert**.

[Table 8-2](#) lists the groups and default permitted actions for user access to MFT embedded server directories. Console access roles and groups do not determine embedded server access.

Table 8-2 MFT Embedded Server Groups and Permissions

Group	Members	Actions Permitted	Additional Notes
Administrators	Any Enterprise user. By default the user named weblogic is a member.	Read, Write, Delete, List (file operations) createDir, renameDir, deleteDir, changeDir (directory operations)	By default all permissions are granted to any member.

Table 8-2 (Cont.) MFT Embedded Server Groups and Permissions

Group	Members	Actions Permitted	Additional Notes
OracleSystemGroup	Any Enterprise user. By default the user named OracleSystemUser is a member.	Read, Write, Delete, List (file operations)	Using this group for access provisioning is not recommended, because this group is intended for internal applications and system management.
Other preexisting groups	Any Enterprise user.	Read, Write, Delete, List (file operations)	Examples of these groups are Monitors and Deployers. These groups and users belonging to them are listed in the Embedded Server User Access tab.
User-created groups	Any Enterprise user.	Read, Write, Delete, List (file operations)	These groups and users belonging to them are listed in the Embedded Server User Access tab.
User-created roles	Any Enterprise user or group.	Based on member groups. If a role has the Administrators group as a member, all operations are allowed. Otherwise only file operations are allowed.	There is no preexisting role for embedded server access provisioning. You can create a role, assign members, and provision access.

Granting Payload Access

You can grant users, groups, and roles access to the payloads of transfers with these characteristics:

- A SOAP, SOA, Service Bus, or ODI target type
- A **Delivery Method** value of Reference
- A **Reference Type** value of FTP

If you grant no specific access, then all users, groups, and roles have access to the transfer payloads.

The steps for this process are:

1. Click the arrow to the left of **Transfers** in the left pane navigator.
The transfers are listed.
2. Click the transfer name or right-click it and then select the Open menu item.
The transfer tab opens.
3. Click the arrow to the left of **Payload Access**.
The Payload Access section opens.
4. Click **add users, groups, and roles**.
The Add Users dialog opens.
5. Select a category: User, Role, or Group.
6. Type part or all of a user, role, or group name in the **Search** field.

You must type at least three letters. Any matches are displayed below the Search field.

7. Select the matching name you want to add.

The selected name appears in the search field.

8. Click **Add to List**.

The name appears in the **Selected Users, Groups, and Roles** list.

To delete a user, role, or group from the list, click the red X to the right of it.

9. Repeat steps 5 through 8 for each user, role, or group you want to add.

10. Click **Add Users**.

To cancel adding users, click **Cancel**.

The Add Users dialog closes.

11. Verify that each user, role, or group you wanted to add is displayed in the Payload Access section of the transfer tab.

To delete a user, role, or group from the list, click the red X to the right of it.

12. **Save** and optionally **Deploy** the transfer.

For more information about configuring transfers, see [Configuring a Transfer](#).

Embedded Server Security

There are two types of Secure embedded servers: sFTP (SSH-FTP) and FTPS (FTP over SSL).

The embedded servers support the following protocols: FTP RFC959, FTP RFC2228, sFTP, FTPS, SSH-2, TLS 1.1, and TLS 1.2. The SSH-1 protocol and SSHD (secure shell) are not supported.

sFTP (SSH-FTP)

[Table 8-3](#) lists the sFTP embedded server settings related to security. These settings are on the Administration page, Embedded Servers tab, and sFTP subtab. After changing any of these settings, you must **Stop** and **Start** the sFTP server to activate the settings.

Table 8-3 sFTP Embedded Server Security Settings

Setting	Description
Authentication Type	Specifies the authentication type: Password (default), Public Key, or Both.
Host Key Alias	Specifies the alias of the SSH private key for authentication. To create SSH keys, see Configuring the SSH Keystore .

FTPS (FTP Over SSL)

[Table 8-4](#) lists the FTP embedded server settings related to security. These settings are on the Administration page, Embedded Servers tab, and FTP subtab. After changing any of these settings, you must **Stop** and **Start** the FTP server to activate the settings.

Table 8-4 FTP Embedded Server Security Settings

Setting	Description
Plain FTP	Enables plain FTP, without Implicit or Explicit SSL support, on the FTP server. You can enable implicit or explicit SSL support, or both, in addition to plain FTP. The default is enabled (checked).
Implicit	Requires the client to immediately challenge the FTPS server with a TLS/SSL <code>ClientHello</code> message. A non-FTPS aware client cannot connect to an implicit SSL-enabled server. The default is enabled (checked).
Explicit	Allows clients to explicitly request that the FTP server encrypt the session and mutually agree to an encryption method. This is known as explicit FTPS or FTPES. Explicit mode is legacy-compatible, so plain FTP clients can still connect to the FTP server. Common commands for invoking FTPS security include <code>AUTH TLS</code> and <code>AUTH SSL</code> . The default is enabled (checked).
Client Authentication	Specifies the level of client authentication: Need, Want, or None. Applies only if Implicit or Explicit is checked. <ul style="list-style-type: none"> • Need - The FTP server's SSL engine <i>requires</i> client authentication during the handshake. • Want - The FTP server's SSL engine <i>requests</i> client authentication during the handshake. • None - No client authentication is performed (default).
Protocol	Specifies the security protocol: TLS (default) or SSL. Applies only if Implicit or Explicit is checked.
Cipher Suite	Specifies the cipher suites to use. To use all available cipher suites, check All. Checking none uses a default list. Applies only if Implicit or Explicit is checked.
Certificate Alias	Specifies the alias of the SSL private key for authentication. Applies only if Implicit or Explicit is checked. To create SSL keys, see Configuring the SSL Keystore .

 **Note:**

The message shown below is not an issue and is information logged to indicate that the FTPS service will not be started when there is no valid port available. This error information is not shown when implicit FTPS service is enabled.

```
{APP: mft-app} [partition-name: DOMAIN]
oracle.mft.COMMON.<MFTServer.initFTPServer>:Invalid value for FTPS
Port: [-1]. FTPS Service will fail to start.
```

Remote SFTP Server Security

The remote SFTP server security algorithms

KeyExchangeAlgorithm

- diffie-hellman-group1-sha1 <!-- deprecated -->
- diffie-hellman-group-exchange-sha1
- diffie-hellman-group14-sha1

- diffie-hellman-group-exchange-sha256
- ecdh-sha2-nistp521
- ecdh-sha2-nistp384
- ecdh-sha2-nistp256

CompressionAlgorithm

- none
- zlib
- zlib@openssh.com

DataIntegrityAlgorithm (MAC)

- hmac-md5 <!-- deprecated -->
- hmac-sha1
- hmac-sha256
- hmac-sha256@ssh.com
- hmac-sha2-512
- hmac-sha2-512-etm@openssh.com
- hmac-sha2-512-96
- hmac-sha2-256-96
- hmac-sha2-256-etm@openssh.com
- hmac-sha2-256
- hmac-ripemd160
- hmac-ripemd160-etm@openssh.com
- hmac-sha1-96
- hmac-sha1-etm@openssh.com

PKIAlgorithm

- ssh-rsa
- ssh-dss <!-- deprecated -->
- x509v3-sign-rsa
- x509v3-sign-rsa-sha1
- ecdsa-sha2-nistp521
- ecdsa-sha2-nistp384
- ecdsa-sha2-nistp256

CipherSuite

- twofish192-cbc <!-- deprecated -->
- cast128-cbc <!-- deprecated -->
- twofish256-cbc <!-- deprecated -->
- aes128-cbc

- aes128-ctr
- twofish128-cbc <!-- deprecated -->
- 3des-cbc
- 3des-ctr
- aes256-cbc
- aes256-ctr
- aes192-cbc
- aes192-ctr

Integrating with Oracle Access Manager 11g for Single Sign-On

You can integrate the Oracle Managed File Transfer console URL with Oracle Access Manager 11g to achieve single-sign-on with other Enterprise web applications.

For general information about installing and configuring Oracle Access Manager, see *Configuring Oracle Access Manager (OAM) in Administering Oracle WebCenter Portal*.

To protect the MFT console URL, follow the steps described in *Configuring the WebLogic Server Administration Console and Enterprise Manager for OAM 11g*, except specify `/mftconsole /mftconsole/* /mftconsole/.../*` as the **Resource URL**.

Message Encryption Using PGP

You can encrypt or decrypt a file to secure data that is being transferred.

For more information, see [Encryption and Decryption at the Source](#) or [Encryption and Decryption Preprocessing](#).

For encryption, you must reference the public PGP key alias. For decryption, you must reference the private PGP key alias.

To set up PGP keys and key aliases in the Oracle Managed File Transfer keystore, see [Configuring the PGP Keystore](#).

FIPS 140 Compliance

FIPS 140-2 specifies the security requirements that must be met by a cryptographic module to protect sensitive information. The standard provides four increasing, qualitative levels of security to cover the wide range of potential applications and environments in which cryptographic modules may be employed.

Oracle Fusion Middleware Release 12c (12.2.1.x) supports the use of FIPS 140-2 enabled cryptographic libraries. The ability to operate in FIPS 140 mode is specific to a defined set of scenarios and transactions supported by Oracle Managed File Transfer. It applies where validated cryptography is used to support or enforce security-sensitive tasks such as authentication, authorization, confidentiality, integrity, and so on.

For more information about FIPS 140–2 use in Oracle Managed File Transfer, see *FIPS 140 Support in Oracle Fusion Middleware* in *Administering Oracle Fusion Middleware*.

In MFT, FIPS 140-2 certification involves certifying:

- Embedded SFTP Server

- Embedded FTP over SSL Server
- PGP Encrypt/Decrypt Actions
- Checksum generation
- SFTPRemote

Enabling FIPS 140–2 Mode from Java Options

To enable FIPS 140-2 mode from Java options, follow these steps:

1. Using the following URL, download and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files that correspond to the version of your JDK. These Java policy JAR files affect cipher key sizes greater than 128 bits.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Open the .ZIP distribution and update `local_policy.jar` and `US_export_policy.jar` in `JAVA_HOME/jre/lib/ security`. See the `README.txt` file in the .ZIP distribution for more information and installation instructions.

2. Create your own `java.security` file. You can use the one that comes with the installed JDK as a guide

Add both the RSA JCE provider and the RSA JSSE provider as the first two Java security providers listed in your `java.security` properties file:

```
#
security.provider.1=com.rsa.jsafe.provider.JsafeJCE
security.provider.2=com.rsa.jsse.JsseProvider

security.provider.3=sun.security.provider.Sun
:
```

3. Start WebLogic Server.

For complete details about enabling FIPS mode in WebLogic Server, see [Enabling FIPS Mode](#)

SFTP

When FIPS is enabled at SFTP server, RSA and DSA keys are not supported. Oracle MFT supports only ECDSA keys of size 256, 384 bits.

Type	Non-FIPS Algorithms	FIPS Algorithms
KeyExchange	DHG1	DHG14
Ciphers	BlowfishCBC	AES128CBC, TripleDESCBC, AES192CBC, AES256CBC
Message Authentication	HMACMD5, HMACMD596, HMACSHA196	HMACSHA1
Signature	RSA, DSA	ECDSA

 **Note:**

When FIPS is enabled at Oracle WebLogic Server, MD5 and Blowfish encryption algorithms are not supported.

FTP-SSL

When FIPS is enabled at FTPS server, keys of size equals to or above 2048 bits are supported. When enabled, by default, the ciphers listed in the Cipher suites list are FIPS compliant. So MFT SSL server supports only those selected in that list.

Any ciphers with MD2, MD4, MD5, RC2, RC4, RC5, DH algorithms in it are not FIPS approved algorithms.

 **Note:**

MFT SSL supports many ciphers in both non-FIPS and FIPS mode. But not all ciphers supported by MFT are supported by the FTP clients. FTP clients may support only subset of the MFT ciphers. Before using any cipher make sure that the FTP client has the support for selected cipher. It is recommended that you use a combination of ciphers which are common across providers along with the specific ones.

PGP

Type	Non-FIPS Algorithm	FIPS Algorithm
Encryption	CAST, BLOWFISH, SAFER, DES, TWOFISH, IDEA (not supported due to licensing)	AES_128, AES_192, AES_256

 **Note:**

In FIPS mode, TRIPLE_DES algorithm is not supported.

Algorithm	MBean Value
NULL	0
IDEA (not supported due to licensing)	1
TRIPLE_DES	2
CAST5	3
BLOWFISH	4
SAFER (not applicable for PGP symmetric keys)	5
DES	6
AES_128	7
AES_192	8
AES_256	9
TWOFISH	10

Checksum Generation

Type	Non-FIPS Algorithm	FIPS Algorithm
Message Digest	MD5	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512



Note:

Only MD5, SHA-1, SHA-256 are available with default provider, that is, in non-FIPS mode.

JCA Transports

Type	Non-FIPS Algorithm	FIPS Algorithm
Data Integrity Algorithm	Hmac-md5	Hmac-sha1, Hmac-sha256, Hmac-sha256@ssh.com
Key Exchange Algorithm	Diffie-hellman-group1-sha1, Diffie-hellman-group-exchange-sha1	Diffie-hellman-group14-sha1
Cipher Suite	Cast128, Twofish256, Twofish128, Blowfish	3Des, Aes128, Aes192, Aes256
PKI Algorithm	DSS	3Des, Aes128, Aes192, Aes256



Note:

When FIPS is enabled at the WebLogic level by adding FIPS jars, MD5/blowfish is not supported even when FIPS at the MFT level is not enabled.

For more information about FIPS 140–2 use in Oracle Managed File Transfer, see *FIPS 140 Support in Oracle Fusion Middleware in Administering Oracle Fusion Middleware*.

Creating an Oracle Managed File Transfer Stripe

The Oracle Managed File Transfer installation MFT keystore stripe is not created by default. You must manually create this stripe using the Oracle Enterprise Manager Fusion Middleware Control console or Oracle WebLogic Scripting Tool (WLST) commands.

- [Using Fusion Middleware Control to Create an Oracle Managed File Transfer Stripe](#)
- [Using WLST Commands to Create an Oracle Managed File Transfer Stripe](#)

Using Fusion Middleware Control to Create an Oracle Managed File Transfer Stripe

The steps to create the MFT keystore stripe using the Oracle Enterprise Manager Fusion Middleware Control console are:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the WebLogic Domain node.
3. Select the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.
For example, the domain might be `soainfra` or `base_domain`.
4. Right-click on the domain and select **Security > Keystore**.
5. Click **Create Stripe**.
6. Enter the name of the new stripe: `mft` (all lowercase).
7. Create a default keystore under the new `mft` stripe named `mftDefaultStore`, with an optional password.
8. Restart the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer. See [Oracle WebLogic Server Startup and Shutdown](#).

Using WLST Commands to Create an Oracle Managed File Transfer Stripe

The steps to create the MFT keystore stripe using WLST commands are:

1. [Start WLST](#).

2. Connect to the Administration Server:

```
connect("username","password","t3://hostname:port")
```

For example:

```
connect("weblogic","weblogic1","t3://localhost:7001")
```

3. Access the Oracle Platform Security Services key store service:

```
svc = getOpssService(name='KeyStoreService')
```

4. Verify if the store is created (`mft/mftDefaultStore`):

```
svc.listKeyStores(appStripe='*')
```

5. If the store is missing, create the SSL keystore called `mft`, and store called `mftDefaultStore`:

```
svc.createKeyStore(appStripe='mft', name='mftDefaultStore', password='P@s$W0rd',  
permission=true)
```

6. Create the SSL keys:

```
svc.generateKeyPair(appStripe='mft', name='mftDefaultStore', password='P@s$W0rd',  
dn='cn=www.mycompany.org', keysize='1024',  
alias='mftssl', keypassword='P@s$W0rd2')
```

7. Restart MFT and test again.

Managing Keystores Using WLST Commands

Oracle Managed File Transfer uses SSL and SSH keys for embedded server security and PGP keys for message encryption. To manage the keystores, you can use the Oracle Managed File Transfer console or Oracle WebLogic Scripting Tool (WLST) commands.

To manage keystores using the Oracle Managed File Transfer console, see [Managing Keystores Using the Oracle Managed File Transfer Console](#).

To manage keystores using Oracle WebLogic Scripting Tool (WLST), see the following sections:

- [Configuring the SSL Keystore](#)
- [Configuring the SSH Keystore](#)
- [Configuring the PGP Keystore](#)

Notes:

- You can use the MFT WLST keystore management commands to generate, import, export, delete, list, and update SSL, PGP, SSH, and PEM keys in the MFT keystore. For more information, see MFT Key Management Commands in *WLST Command Reference for SOA Suite*.
- SSL keys in binary (DER) format are not supported. Use keys in BASE64 (PEM or CER) format. You can convert key formats using the `openssl` command.

Key lengths greater than 1024 bit are supported. However, there are some export restrictions on key lengths greater than 1024 bit. These restrictions are mostly specified at the JRE level in the `JAVA_HOME\jre7\lib\security` directory.
- To create additional keystores, you can use WLST commands or the Oracle Enterprise Manager Fusion Middleware Control console. See Managing Keys and Certificates with the Keystore Service in *Securing Applications with Oracle Platform Security Services*, Managing Keystores, Wallets, and Certificates in *Administering Oracle Fusion Middleware*, and Managing the Credential Store in *Securing Applications with Oracle Platform Security Services* for more information about Fusion Middleware Control.

Configuring the SSL Keystore

The default keystore is used for storing Oracle Managed File Transfer SSL keys and certificates. To configure the default keystore, use WLST and the Oracle Managed File Transfer console.

The steps for this process are:

1. [Start WLST](#).
2. Access the Oracle Platform Security Services key store service:

```
svc = getOpssService(name='KeyStoreService')
```

3. Create the SSL keystore:

```
svc.createKeyStore(appStripe='StripeName', name='StoreName',  
password='StorePassword', permission=false/true)
```

For example:

```
svc.createKeyStore(appStripe='mft', name='mftDefaultStore', password='P@s$W0rd',  
permission=true)
```

4. Create the SSL keys:

```
svc.generateKeyPair(appStripe='StripeName', name='StoreName',  
password='StorePassword', dn='cn=CompanyURL', keysize='1024',  
alias='Alias', keypassword='KeyPassword')
```

For example:

```
svc.generateKeyPair(appStripe='mft', name='mftDefaultStore', password='P@s$W0rd',
dn='cn=www.mycompany.org', keysize='1024',
alias='mftssl', keypassword='P@s$W0rd2')
```

Specify `mft` as the stripe name and `mftDefaultStore` as the store name. Oracle Managed File Transfer uses these names by default. The store and key passwords are optional. See [Using Fusion Middleware Control to Create an Oracle Managed File Transfer Stripe](#).

When securing the FTP server, you reference the SSL private key alias configured in this step. See the **Certificate Alias** description in [FTPS \(FTP Over SSL\)](#).

5. [Exit WLST](#).
6. In the Oracle Managed File Transfer console, on the left pane of the Administration page, click **Keystores**.
7. If you specified key and keystore passwords in previous steps, enter the SSL key password in the **Private Key Password** field and the keystore password in the **Keystore Password** field.
8. Click **Save**.

Configuring the SSH Keystore

To configure the SSH keystore, use WLST and the Oracle Managed File Transfer console.

The steps for this process are:

1. [Start WLST](#).
2. Use the `generateKeys` WLST command to create a password-protected private SSH key. The key type is RSA and the key size is 1024 bits. For example:

```
generateKeys('SSH', 'P@s$W0rd', '/export/ssh/ssh-pvt-keys.ppk')
```

Note:

To generate a private RSA key of PEM format, which is used to connect to Oracle Cloud Infrastructure when the OCI Storage Cloud Service type is selected as a source or target, you cannot use the Oracle Managed File Transfer console or the WLST `generateKeys` command. Instead, you can use an external key generation application, such as `ssh-keygen`, or follow the steps in [How to Generate an API Signing Key](#) in the Oracle Cloud Infrastructure documentation. Then, you can import the RSA key of PEM format.

If you are an advanced user and want to set additional key parameters, you can use the `ssh-keygen` command. For example:

```
ssh-keygen -t rsa -b 2048 -f /export/ssh/ssh-pvt-keys.ppk -N P@a$W0rd
```

For more information about `ssh-keygen`, see <http://linux.die.net/man/1/ssh-keygen>.

The password is optional for either command.

3. Use the `importCSFKey` WLST command to import and create an alias for the key.

When securing the sFTP server, you reference the private key alias configured in this step. See the **Host Key Alias** description in [sFTP \(SSH-FTP\)](#).

For example:

```
importCSFKey('SSH', 'PRIVATE', 'mftssh', '/export/ssh/ssh-pvt-keys.ppk')  
importCSFKey('PEM', 'PRIVATE', 'mftpem', '/export/pem/pem-pvt-keys.pem')
```

 **Note:**

An RSA key of PEM format is used to connect to Oracle Cloud Infrastructure when the OCI Storage Cloud Service type is selected as a source or target.

4. [Exit WLST](#).
5. In the Oracle Managed File Transfer console, on the left pane of the Administration page, click **Keystores**.
6. If you specified a password in step 2, enter the SSH key password in the **Private Key Password** field.
7. Click **Save**.

Configuring the PGP Keystore

To configure the PGP keystore, use WLST and the Oracle Managed File Transfer console.

 **Note:**

If a payload is encrypted by a PGP tool outside of MFT using a key length or algorithm that is restricted, MFT decryption will fail. These restrictions are mostly specified at the JRE level in the `JAVA_HOME\jre7\lib\security` directory.

The steps for this process are:

1. [Start WLST](#).
2. Use the `generateKeys` WLST command to create a password-protected PGP key pair. For example:

```
generateKeys('PGP', 'P@s$W0rd', '/export/pgp', 'example<example@example.com>')
```

The password is optional.

3. Use the `importCSFKey` WLST command to import and create an alias for each key. For example:

```
importCSFKey('PGP', 'PUBLIC', 'mftpgppub', '/export/pgp/pub.asc')  
importCSFKey('PGP', 'PRIVATE', 'mftpgppri', '/export/pgp/secret.asc')
```

For encryption, you must reference the public PGP key alias. For decryption, you must reference the private PGP key alias. For more information, see [Encryption and Decryption at the Source](#) or [Encryption and Decryption Preprocessing Actions](#).

4. [Exit WLST](#).
5. In the Oracle Managed File Transfer console, on the left pane of the Administration page, click **Keystores**.

6. If you specified a password in step 2, enter the PGP key password in the **Private Key Password** field.

 **Note:**

The PGP Keystore supports only one private key password; you may have multiple private keys but the password to the private keys must be the same.

7. Click **Save**.

Enable Private Key Passwords for PGP Keys

Perform the following steps to add a distinct passphrase for each PGP key and configure it to override the PGP keystore's passphrase for that particular artifact (source or target).

1. Log in to your Oracle Managed File Transfer instance.
2. In the Oracle Managed File Transfer homepage, navigate to the **Design** tab and click the arrow to the left of **Sources** in the left pane navigator.

The sources are listed.

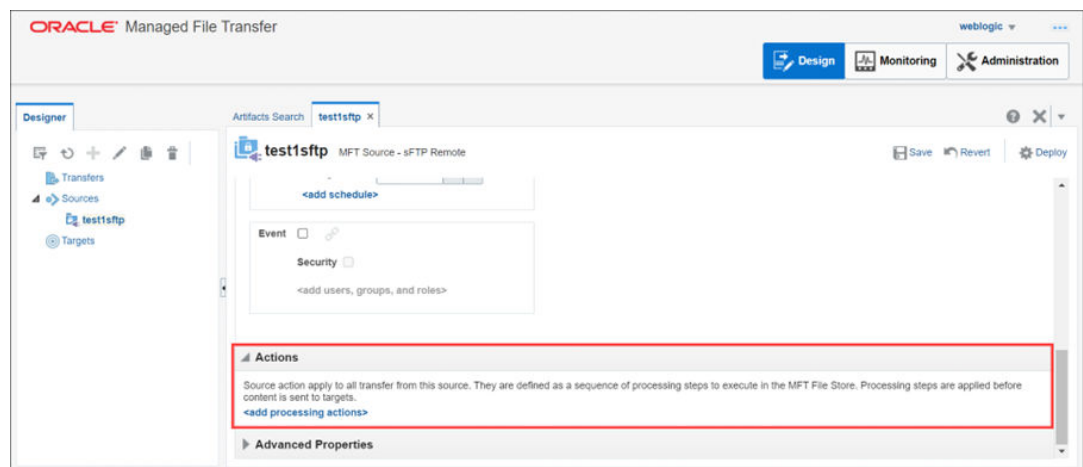
3. Click the source name or right-click and then select the **Open** menu item.

The source tab opens.

4. Click the arrow to the left of **Actions**.

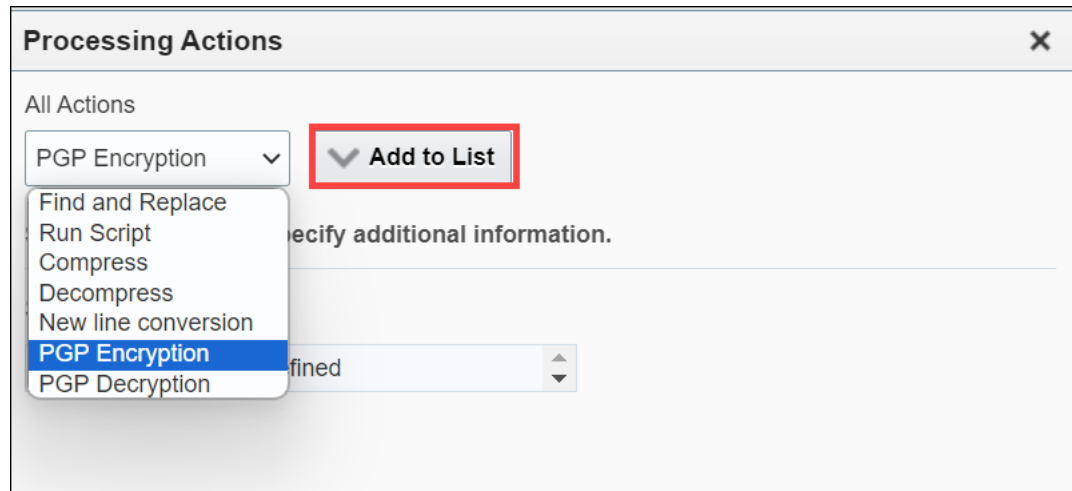
The Actions section opens.

5. Click **add processing actions**.



The Processing Actions dialog opens.

6. Select **PGP Encryption** or **PGP Decryption** from the **All Actions** drop-down list and click **Add to List**.



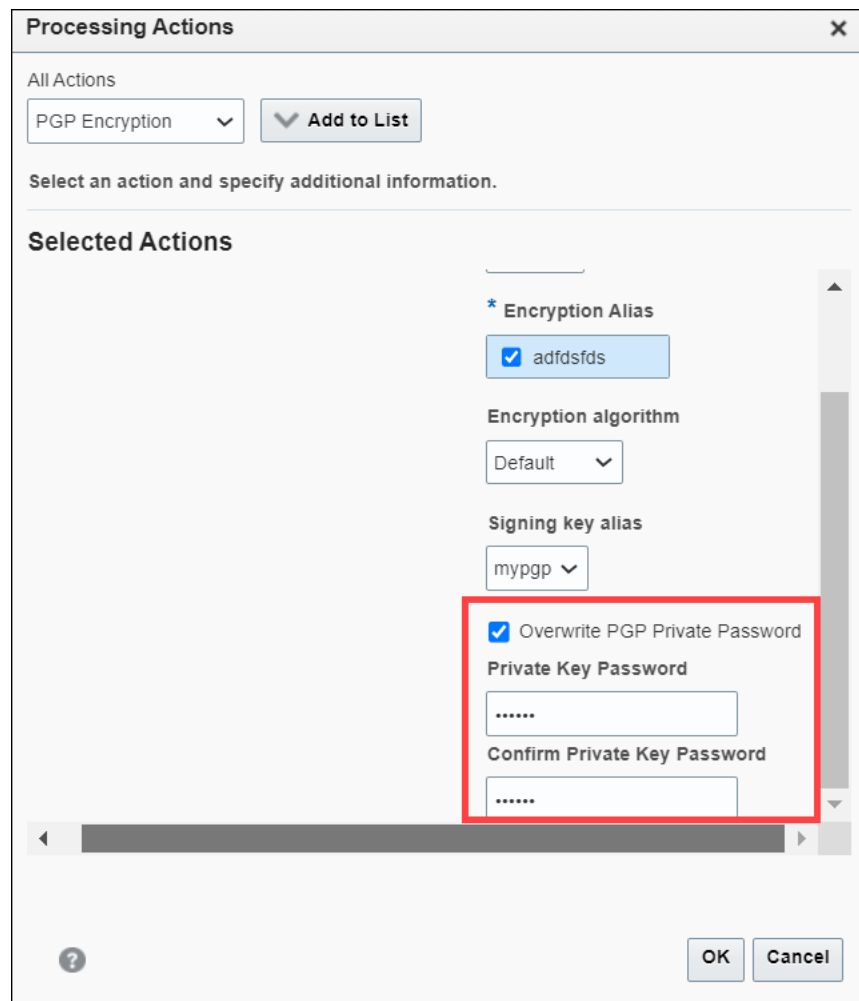
- a. If you selected PGP Encryption, select the values from the Encryption Alias, Armored, Encryption algorithm, and Signing key alias drop-down lists.
 - **Encryption Alias:** the public key alias for encryption. For more information about the key aliases, see [Configuring the PGP Keystore](#).
 - **Armored:** Binary or ASCII. Use ASCII if non-printing characters might be stripped in transit.
 - **Encryption algorithm:** Select from the following supported algorithms.

 **Note:**

If no algorithm is selected, global algorithm settings will apply.

- Default
- Triple-DES
- CAST5 – set as default algorithm
- Blowfish
- DES
- AES-128
- AES-192
- AES-256 – set as default algorithm if FIPS mode is enabled
- Twofish

- **Signing key alias:** Select from the list of imported private signing keys. The Overwrite PGP Private Password checkbox is displayed when you select a private PGP key as signing key alias.



Enable the **Overwrite PGP Private Password** checkbox if the signing key alias is password protected (during the PGP key generation process) and the password differs from the one stored in the PGP keystore.

- b. If you selected PGP Decryption, select the decryption alias from the drop-down list, which is the private key alias for decryption. The decryption alias must already be imported into the PGP keystore.

For more information about the key aliases, see [Configuring the PGP Keystore](#).

The Overwrite PGP Private Password checkbox is displayed when you select a private PGP key as decryption alias.

Enable the **Overwrite PGP Private Password** checkbox if the decryption alias is password-protected (during the PGP key generation process) and the password differs from the one stored in the PGP keystore.

Once enabled, you can add a **Private Key Password** for the artifact. The private key password overrides the existing PGP keystore's passphrase for that particular artifact (source or target).

7. Click **OK**.
To cancel adding actions, click **Cancel**.
8. **Save** and optionally **Deploy** the source.

After successful transfer, you can monitor the result in the Monitor dashboard. See [Monitoring Deployed Sources, Targets, and Transfers](#) and [Transfer Reports](#).

Enabling Security Audit Logging

You can enable audit logging for Oracle Managed File Transfer using Oracle Enterprise Manager Fusion Middleware Control or the Oracle WebLogic Scripting Tool (WLST).

To generate reports of audit data, see [Using Audit Analysis and Reporting](#) in *Securing Applications with Oracle Platform Security Services*.

Using Fusion Middleware Control to Enable Audit Logging

The steps for this process are:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Weblogic Domain node.
3. Select the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.

For example, the domain might be `soainfra` or `base_domain`.

4. Right-click on the domain and select **Security > Audit Policy**.
5. Select `MFT` from the **Audit Component Name** drop-down list.

If you do not see the MFT component, then restart the Oracle WebLogic Server admin server and managed servers. For more information, see [Oracle WebLogic Server Startup and Shutdown](#).

6. Change the **Audit Level** to `Medium`.
7. Click **Apply**.
8. Restart the Oracle WebLogic Server admin server and managed servers. See [Oracle WebLogic Server Startup and Shutdown](#).

Using WLST to Enable Audit Logging

The steps for this process are:

1. Start WLST using the steps described in [Running WLST Commands](#).
2. Use the following WLST command to enable audit logging for MFT:

```
setAuditPolicy(componentType="MFT",filterPreset="None")
```

You can specify a `filterPreset` of `Custom` or `Medium` instead of `None`.

3. Exit WLST using the steps described in [Running WLST Commands](#).

For more information about WLST commands for audit policies, see [Manage Audit Policies with WLST](#) in *Securing Applications with Oracle Platform Security Services*.

FTP/SFTP Operation in Audit Report

FTP/SFTP operations are recorded as part of an audit report. The attributes in the report are User, Command, Server Type (FTP or SFTP), File/Directory, Host IP, and Host Name.

FTP Operation

- PUT
- GET
- APPEND
- RENAME
- DELETE
- LIST
- CWD
- RENAMEDIR
- MKDIR
- RMDIR

SFTP Operation

- PUT
- GET
- DELETE
- RENAME
- LIST
- CWD
- RENAMEDIR
- RMDIR
- MKDIR



Note:

Append is not supported as a separate command for SFTP.

OWSM Security Policy Attachment

Oracle Managed File Transfer supports securing web service sources and targets with Oracle Web Services Manager (OWSM) policies. Web service sources and targets include those of type SOAP, SOA, Service Bus, and ODI.

**Note:**

WS-Security compliant policies are not supported.

You can attach one policy file per source or target. The policy file holds the attached policies and overridden attributes.

You can attach policies globally using Oracle Enterprise Manager Fusion Middleware Control or the Oracle WebLogic Scripting Tool (WLST). You can attach policies locally using the MFT console or WLST.

Web services security can be divided into the following parts:

- Design time — when policies are attached and registered for the source or target.
- Runtime — when the policies are enforced upon invoking the secured source or target.
- Life cycle — how policies are managed regarding the life cycle of the source or target.

This section includes the following topics:

- [Using Fusion Middleware Control for Global Policy Attachment](#)
- [Using WLST for Global Policy Attachment](#)
- [Using the MFT Console for Local Policy Attachment](#)
- [Using WLST for Local Policy Attachment](#)
- [How the Policy Is Applied at Runtime](#)
- [Policies and Artifact Life Cycle Management](#)
- [Verifying Policy Registration](#)

Using Fusion Middleware Control for Global Policy Attachment

How you attach a policy depends on whether the policy is inbound (for sources) or outbound (for targets). In addition, some client policies require credentials.

Managing Policy Credentials

Some client policies, such as the user name token policy, require credentials. Before you can attach such a policy using Oracle Enterprise Manager Fusion Middleware Control, you must create a map and key. For more information, see *Managing the Credential Store in Securing Applications with Oracle Platform Security Services*.

The steps for this process are:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Weblogic Domain node.
3. Select the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.

For example, the domain might be `soainfra` or `base_domain`.

4. Right-click on the domain and select **Security > Credentials**.

The Credentials table appears.

5. Click **Create Map**.
6. In the Create Map dialog, type `oracle.wsm.security` in the **Map Name** field. Click **OK**.
The `oracle.wsm.security` credential appears in the Credentials table.
7. Select the `oracle.wsm.security` credential and click **Create Key**.
8. In the Create Key dialog, type `basic.credentials` in the **Key** field. Type the user information in the **User Name**, **Password**, and **Confirm Password** fields. Click **OK**.
An expand arrow appears to the left of the `oracle.wsm.security` credential. Clicking this arrow displays the `basic.credentials` key.

Creating a Policy Set for a Source

You can create a policy set and attach a policy for a source. For more information, see *Managing Web Service Policies with Fusion Middleware Control* in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

The steps for this process are:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Weblogic Domain node.
3. Select the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.
For example, the domain might be `soainfra` or `base_domain`.
4. Right-click on the domain and select **Web Services > WSM Policy Sets**.
The WSM Policy Set Summary table appears.
5. Click **Create**.
6. On the Create Policy Set: Enter General Information page, enter a **Name**, check **Enabled**, and select **SOAP Web Service** from the Type of Resource drop-down list. Click **Next**.
7. On the Create Policy Set: Enter Resource Scope page, type the following information:
 - **Domain Name:** Type the name of the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.
For example, the domain might be `soainfra` or `base_domain`.
 - **Application Name:** Type `*. or mftapp`
 - **Application Module Name or Connection Name:** Type `*. or mftapp`
 - **RESTful Application, Service, or Web Service Endpoint Name:** This can be a SOA composite endpoint or an Oracle Managed File Transfer source. For a source, type `{http://xmlns.oracle.com/fmw/mft/soap}MFTService_source-name`.
 - **Port Name:** Type `MFTServicePort` or the port name of a SOA composite (for example `submit_ptt`).
8. Click **Next**. On the Create Policy Set: Enter Constraint page, click **Next** again.
9. On the Create Policy Set: Add Policy References page, select one or more policies from the Available Policies list and click **Attach**. Click **Next**.
10. On the Create Policy Set: Summary page, click **Save**.

Creating a Policy Set for a Target

You can create a policy set and attach a policy for a target. For more information, see *Managing Web Service Policies with Fusion Middleware Control in [Securing Web Services and Managing Policies with Oracle Web Services Manager](#)*.

The steps for this process are:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Weblogic Domain node.
3. Select the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.
For example, the domain might be `soainfra` or `base_domain`.
4. Right-click on the domain and select **Web Services > WSM Policy Sets**.
The WSM Policy Set Summary table appears.
5. Click **Create**.
6. On the Create Policy Set: Enter General Information page, enter a **Name**, check **Enabled**, and select **SOAP Web Service Client** from the Type of Resource drop-down list. Click **Next**.
7. On the Create Policy Set: Enter Resource Scope page, type the following information:
 - **Domain Name:** Type the name of the domain on which the Oracle WebLogic Server managed server dedicated to Oracle Managed File Transfer is installed.
For example, the domain might be `soainfra` or `base_domain`.
 - **Application Name:** Type *
 - **Application Module Name or Connection Name:** Type *
 - **RESTful Application, Service, or Web Service Endpoint Name:** Specify a SOA composite endpoint name.
 - **Port Name:** Port Name of the target SOA composite.
8. Click **Next**. On the Create Policy Set: Enter Constraint page, click **Next** again.
9. On the Create Policy Set: Add Policy References page, select one or more policies from the Available Policies list and click **Attach**. Click **Next**.
10. On the Create Policy Set: Summary page, click **Save**.

Using WLST for Global Policy Attachment

Using WLST, you can attach a policy to all web service endpoints in Oracle Managed File Transfer.

The steps for this process are:

1. Start WLST using the steps described in [Running WLST Commands](#).
2. Use the following WLST commands to create and attach a policy set:

```
beginRepositorySession()  
createPolicySet('mft', 'ws-service', 'Domain("*")')  
attachPolicySetPolicy('oracle/wss_username_token_service_policy')  
validatePolicySet()
```

```
commitRepositorySession()
displayPolicySet('mft')
```

The `wss_username_token_service_policy` is an example. You can attach a different policy.

3. Exit WLST using the steps described in [Running WLST Commands](#).

You can also attach a policy to a specific web service endpoint, either a source or target.

The steps for this process are:

1. Start WLST using the steps described in [Running WLST Commands](#).
2. Use the following WLST commands to create and attach a policy set:

```
beginRepositorySession()
createPolicySet('mft', 'ws-service', 'Service("{http://xmlns.oracle.com/fmw/mft/soap}MFTService_SOAPSource"')
attachPolicySetPolicy('oracle/wss_username_token_service_policy')
validatePolicySet()
commitRepositorySession()
displayPolicySet('mft')
```

The `SOAPSource` is an example. You can create a policy set for a different source or target.

The `wss_username_token_service_policy` is an example. You can attach a different policy.

3. Exit WLST using the steps described in [Running WLST Commands](#).

For more information about WLST commands for managing policy sets, see *Policy Set Management Commands* in *WLST Command Reference for Infrastructure Components*.

Using the MFT Console for Local Policy Attachment

The steps for this process are:

1. Create and deploy the SOAP, SOA, Service Bus, or ODI application corresponding to the MFT source or target. For a target, you must attach a policy to the application.

See [Integrating Oracle Managed File Transfer with Other Products](#) for more information.

2. Create or open the source or target as described in [Creating a Source](#) or [Creating a Target](#).

3. Before clicking **Save**, click the arrow to the left of Policies.

The Selected Policies and Available Policies tables are displayed.

4. To search the Available Policies by name, type a full or partial name in the **Search** field and click the **Search** icon.

5. Select one or more policies from the Available Policies list.

6. Click **Attach**.

The selected policies move to the Selected Policies list.

To detach one or more policies, select them and click **Detach**.

7. Click **Save**.
8. Add the source or target to a transfer as described in [Configuring a Transfer](#).
9. Deploy the transfer as described in [Deploying and Testing Transfers](#).

The attached policies are automatically registered in OWSM. To verify the registration, see [Verifying Policy Registration](#).

Using WLST for Local Policy Attachment

You can attach a policy to a specific web service endpoint, either a source or target.

The steps for attaching a policy are:

1. Start WLST using the steps described in [Running WLST Commands](#).
2. Use the following WLST commands to create and attach a policy set:

```
beginWSMSession()
listWSMPolicySubjects('mft-app')
selectWSMPolicySubject('/weblogic/soainfra/mft-app', '#SOAPSource',
'WSService({http://xmlns.oracle.com/fmw/mft/
soap}MFTService_SOAPSource#MFTServicePort)')
attachWSMPolicy('oracle/binding_authorization_denyall_policy')
attachWSMPolicy('oracle/wss_username_token_service_policy')
previewWSMEffectivePolicySet()
commitWSMSession()
```

The `SOAPSource` is an example. You can create a policy set for a different source or target.

The `wss_username_token_service_policy` is an example. You can attach a different policy.

3. Exit WLST using the steps described in [Running WLST Commands](#).

The steps for detaching a policy are:

1. Start WLST using the steps described in [Running WLST Commands](#).
2. Use the following WLST commands to create and attach a policy set:

```
beginWSMSession()
listWSMPolicySubjects('mft-app')
selectWSMPolicySubject('/weblogic/soainfra/mft-app', '#SOAPSource',
'WSService({http://xmlns.oracle.com/fmw/mft/
soap}MFTService_SOAPSource#MFTServicePort)')
detachWSMPolicy('oracle/binding_authorization_denyall_policy')
detachWSMPolicy('oracle/wss_username_token_service_policy')
previewWSMEffectivePolicySet()
commitWSMSession()
```

The `SOAPSource` is an example. You can create a policy set for a different source or target.

The `wss_username_token_service_policy` is an example. You can attach a different policy.

3. Exit WLST using the steps described in [Running WLST Commands](#).

For more information about WLST commands for managing policy sets, see [Policy Set Management Commands in *WLST Command Reference for Infrastructure Components*](#).

How the Policy Is Applied at Runtime

For a source, the sending user must specify the policy and its required credentials with the file to be transferred. Otherwise the file transfer will not succeed.

For a target, the receiving user does not need to specify the policy. This is the responsibility of the sending user in the transfer. However, if the sending user does not specify the policy attached to the target and its required credentials, the file transfer will not succeed.

Credentials that the policy requires can include a username, password, certificate, or other security information. See [Managing Policy Credentials](#) for more information.

Policies and Artifact Life Cycle Management

A policy file is persisted in the MFT metadata store (MDS) and follows the exact life cycle pattern of its parent source or target artifact, including the version.

- Create an artifact, and the policy file is created in the MDS and referenced by the artifact.
- Deploy an artifact, and the policy file is automatically registered in OWSM.
- Undeploy an artifact, and the policy file is de-registered.
- Enable an artifact, and the policy file is automatically registered in OWSM.
- Disable an artifact, and the policy file is de-registered.
- Delete an artifact, and the policy file is deleted from the MDS.
- Export an artifact, and the policy file is exported and linked to by the artifact export file.
- Import an artifact, and the linked policy file is also imported.

Verifying Policy Registration

After you attach policies globally or locally, you can use WLST to verify that these policies are registered in the OWSM repository as part of the effective policy set.

The steps for this process are:

1. Start WLST using the steps described in [Running WLST Commands](#).
2. Use the following WLST commands to specify the MFT policy subject and display the effective policy set:

```
beginWSMSession()
listWSMPolicySubjects('mft-app')
selectWSMPolicySubject('/weblogic/soainfra/mft-app', '*', '*')
displayWSMEffectivePolicySet()
commitWSMSession()
```

The output of the `displayWSMEffectivePolicySet` command should list the policies you attached.

3. Exit WLST using the steps described in [Running WLST Commands](#).

For more information about the WLST commands for policy subjects, see Policy Subject Commands in *WLST Command Reference for Infrastructure Components*.

For general information about the OWSM repository, see Overview of Web Services Administration in *Administering Web Services*.

Configuring SSL only Domain for Oracle Managed File Transfer

You can configure a SSL only domain in Oracle Managed File Transfer.

To configure SSL only domain:

1. Go to `$DOMAIN_HOME/oracle_common/bin/` directory and run the following command:

```
sh libovdconfig.sh -host <host> -port <port> -userName <wlsadminusername> -
domainPath <AbsolutePathOfDomainHome> -createKeystore
```

2. Extract the certificate of AdminServer by connecting to `http://host:port/console` from the web browser and export the certificate to a file. Enter the certificate details.

The file format should be "Base 64 Encoded x.509".

 **Note:**

The above step is applicable only for integrated Weblogic LDAP (Default Authenticator). For other LDAPs, the certificate has to be exported by using the appropriate LDAP commands.

3. Import the certificate exported in the above step to the created keystore by executing the following command.

```
keytool -importcert -keystore <DOMAIN_HOME>/config/fmwconfig/ovd/default/
keystores/adapters.jks -storepass <password> -alias <alias> -file <filePath> -
noprompt
```

The password is specified while generating the keystore and filepath is the file containing the exported certificate.

4. Choose any alias of your choice.
5. Perform the SSL related changes to create SSL only domain.
6. Restart the Weblogic Admin and Managed Servers.

After restarting the server, you will be able to login to MFT console and embedded servers in SSL only domain.

After creating the domain, you need to enable the SSL domain. To enable SSL domain, see [Enabling SSL only Domain](#).

Enabling SSL only Domain

After creating the SSL only domain, enable the domain by configuring SSL in different use cases by following the steps below:

Enabling SSL only Domain

Follow the steps below to set up Oracle MFT in an SSL only WebLogic domain.

1. Enable the SSL listening port at the time of domain creation, in case of new domains.
2. For existing domain, extend the domain and select the SSL listening port.
3. Start the AdminServer and disable the non-SSL listening port of the managed Server and AdminServer.
4. Start the managed server.

Invoking Webservice over SSL (HTTPS service)

Follow the steps below to invoke the SOA/SOAP targets over HTTP SSL protocol.

1. Export the SSL certificate of remote SSL webservice, for example, from Web browser to File.

2. Import the certificate to below stores:
 - a. `jdk_home/jre/lib/security/cacerts`
 - b. `/wlserver/server/lib/cacerts`
 - c. EM > Security > Keystore > System > Trust store, CA store
3. Restart the server.

Configuring remote SSL FTP server

Follow the steps below to connect from a Remote FTP(S) Source or Target to a remote FTP Server. The remote FTP Server can be non-MFT FTP Server, or it could be MFT Embedded FTP Server within another deployment of MFT.

1. Export the trusted certificate used for remote SSL FTP server to a file using appropriate tools. In case of MFT FTPS remote server, do it via WLST with below command:

```
svc.exportKeyStoreCertificate(appStripe='<stripe>', name='<store>', password='<storePassword>', alias='<certalias>', type='Certificate', filepath='<filePath>')
```

2. Import the trusted certificate to MFT using the below WLST command:

```
svc.importKeyStoreCertificate(appStripe='<Stripe>', name='<Store>', password='<StorePassword>', alias='<certAlias>', keypassword='<keyPassword>', type='TrustedCertificate', filepath='<filepath>')
```

9

Using WLST Commands with Oracle Managed File Transfer

Learn how to use the WLST (Oracle WebLogic Scripting Tool) commands that perform Oracle Managed File Transfer (MFT) operations.

This chapter includes the following sections:

- [Running WLST Commands](#)
- [MFT WLST Command Summary](#)
- [Oracle MFT EJBs](#)

For more detailed descriptions and examples of the MFT WLST commands, see the Oracle Managed File Transfer Custom WLST Commands in *WLST Command Reference for SOA Suite*.

Running WLST Commands

Before you can run WLST commands, you must start WLST and connect to the Oracle WebLogic Server Managed Server dedicated to Oracle Managed File Transfer.

The steps for this process are:

1. Go to the Oracle WebLogic Server command directory for MFT:

```
cd ${MW_HOME}/mft/common/bin
```

2. Run the Oracle WebLogic Scripting Tool:

```
./wlst.sh
```

3. Connect to the Oracle WebLogic Server Managed Server dedicated to MFT:

```
connect("username","password","t3://hostname:port")
```

For example:

```
connect("weblogic","weblogic1","t3://localhost:7011")
```

4. Run WLST commands as needed.

Use this command to list MFT WLST commands:

```
help("mft")
```

Use this command to list short names for MFT WLST commands:

```
help("mft-shortcuts")
```

The commands work in the same way whether you use the long or short names.

5. Disconnect and exit.

```
disconnect()  
exit()
```

MFT WLST Command Summary

Use WLST (Oracle WebLogic Scripting Tool) commands to perform various Oracle Managed File Transfer operations.

Table 9-1 summarizes the MFT WLST commands. It is intended as a quick reference and not as a complete description of each command. For a complete description of these commands, see Oracle Managed File Transfer Custom WLST Commands in *WLST Command Reference for SOA Suite*.

Table 9-1 MFT WLST Command Summary

Command	Shortcut	Syntax	Description
bulkDeployArtifact	buDepAF	bulkDeployArtifact('TRANSFER SOURCE TARGET', 'artifact_name', 'comment')	Deploys a comma-separated list of source, transfer, or target artifacts or * for all. A comment is optional.
createMftCredential	N/A	createMftCredential(password, key)	Creates the credential for mftapp. Enter the password for which the credential needs to be created and the key for the credential.
deleteArtifact	delAF	deleteArtifact('TRANSFER SOURCE TARGET', 'artifact_name')	Deletes a source, transfer, or target artifact.
deleteArtifactDeployment	delDepAF	deleteArtifactDeployment('TRANSFER SOURCE TARGET', 'artifact_name', 'label')	Deletes an undeployed source, transfer, or target artifact. Also use to delete old configuration history. Use Show Deployment Details on the Deployment tab to view the <i>label</i> .
deployArtifact	depAF	deployArtifact('TRANSFER SOURCE TARGET', 'artifact_name', 'comment')	Deploys a source, transfer, or target artifact. A comment is optional.
disableArtifact	disAF	disableArtifact('TRANSFER SOURCE TARGET', 'artifact_name', 'comment')	Disables a deployed and previously enabled source, transfer, or target artifact. A comment is optional.
enableArtifact	enAF	enableArtifact('TRANSFER SOURCE TARGET', 'artifact_name', 'comment')	Enables a deployed and previously disabled source, transfer, or target artifact. A comment is optional.

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
exportDeployedArtifact	expDepAF	<pre>exportDeployedArtifact('artifact_type','artifact_name',label,'archive_file_path', generate_config_plan, long_format)</pre>	<p>Exports a deployed source, transfer, or target artifact to a ZIP file. Use Show Deployment Details on the Deployment tab to view the <i>label</i>. If you are connecting to WLST remotely, the ZIP file is created on the remote server.</p> <p><i>generate_config_plan</i> (optional): Indicates whether to generate the mftConfig XML. Config plan is generated in same folder where archive file is generated. Default is <code>FALSE</code>.</p> <p><i>long_format</i> (optional): If <code>TRUE</code>, most of the attributes will be included in the config plan xml; otherwise, only the key attributes will be listed in the config plan XML. Default is <code>FALSE</code>.</p>
isArtifactInMDS	isAFinMDS	<pre>isArtifactInMDS('TRANSFER SOURCE TARGET', 'artifact_name')</pre>	<p>Checks whether a source, transfer, or target artifact exists in the MDS (Metadata Store) and returns <code>TRUE</code> or <code>FALSE</code>.</p>
undeployArtifact	undepAF	<pre>undeployArtifact('TRANSFER SOURCE TARGET', 'artifact_name', 'comment')</pre>	<p>Undeploys a source, transfer, or target artifact without deleting it from the configuration. A comment is optional.</p>
exportMftMetadata	expMD	<pre>exportMetadata('archive_file', generate_config_plan, long_format)</pre>	<p>Exports the entire MFT configuration, excluding passwords, to a ZIP file. If you are connecting to WLST remotely, the ZIP file is created on the remote server.</p> <p><i>generate_config_plan</i> (optional): Indicates whether to generate the mftConfig XML. Config plan is generated in same folder where archive file is generated. Default is <code>FALSE</code>.</p> <p><i>long_format</i> (optional): If <code>TRUE</code>, most of the attributes will be included in the config plan xml; otherwise, only the key attributes will be listed in the config plan XML. Default is <code>FALSE</code>.</p>

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
exportTransferMetadata	expXfrMD	<code>exportTransferMetadata('archive_file', 'transfer_name', generate_config_plan, long_format)</code>	Exports a transfer artifact and related metadata to a ZIP file. If you are connecting to WLST remotely, the ZIP file is created on the remote server. <i>generate_config_plan</i> (optional): Indicates whether to generate the mftConfig XML. Config plan is generated in same folder where archive file is generated. Default is FALSE. <i>long_format</i> (optional): If TRUE, most of the attributes will be included in the config plan xml; otherwise, only the key attributes will be listed in the config plan XML. Default is FALSE.
importMftMetadata	impMD	<code>importMetadata('archive_file', generate_config_plan, previewMode)</code>	Imports a previously exported MFT configuration from a ZIP file. <i>generate_config_plan</i> (optional): Indicates whether to generate the mftConfig XML. Config plan is generated in same folder where archive file is generated. Default is FALSE.
resetMetadata	resMD	<code>resetMetadata('preserve_references')</code>	Resets the MFT configuration, deleting all artifacts and resetting all administrative settings to their defaults. Example: <pre>MW_HOME/oracle_common/ common/bin/wlst.sh connect("weblogic","weblogic1","t3:// mftserver:mftport") resetMetadata(FALSE)</pre>
deleteCSFKey	delKey	<code>deleteCSFKey('SSH PGP PEM', 'PRIVATE PUBLIC', 'alias')¹</code>	Deletes a key alias from the MFT keystore. Example: <pre>deleteCSFKey('SSH', 'PRIVATE', 'my-alias')</pre>

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
exportCSFKey	expKey	exportCSFKey('SSH PGP PEM', 'PRIVATE PUBLIC', 'zip_file_path') ¹	<p>Exports keys from the MFT keystore to a zip file containing the keys.</p> <p>Example:</p> <pre>exportCSFKey('SSH', 'PRIVATE', '/export/ssh/ my_private_keys.zip')</pre> <p>Unzip the file to extract the keys.</p>
generateKeys	genKeys	generateKeys('SSH PGP', 'password', 'key_file_path')	<p>Generates keys and saves them to one or more key files. The key type is RSA and the key size is 1024 bits. The private key password is optional.</p> <p>For SSH, the path must include the key file name.</p> <p>For PGP, two files are generated under the specified path: the <code>secret.asc</code> file contains the PGP private key, and the <code>pub.asc</code> file contains the PGP public key.</p> <p>To generate a private RSA key of PEM format, which is used to connect to Oracle Cloud Infrastructure when the OCI Storage Cloud Service type is selected as a source or target, you cannot use the Oracle Managed File Transfer console or the WSLT <code>generateKeys</code> command. Instead, you can use an external key generation application, such as <code>ssh-keygen</code>, or follow the steps in How to Generate an API Signing Key in the Oracle Cloud Infrastructure documentation. Then, you can import the RSA key of PEM format.</p> <p>¹</p> <p>Example:</p> <pre>generateKeys('SSH', '', '/ export/ssh/ssh-pvt- keys.ppk')</pre>

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
importCSFKey	impKey	importCSFKey('SSH PGP PEM', 'PRIVATE PUBLIC', 'alias', 'key_file_path') ¹	Imports a key to the MFT keystore from a key file and creates an alias. Example: importCSFKey('SSH', 'PRIVATE', 'my-alias', '/export/ssh/my_private_keys.ppk' Before you can use the OCI Storage Cloud Service type as a source or target, you must import RSA key of PEM format to connect to Oracle Cloud Infrastructure. ¹ Example: importCSFKey(PEM, 'PRIVATE', 'my-alias', '/export/pem/my_private_keys.pem
listCSFKeyAliases	lsKeyAliases	listCSFKeyAliases('SSH PGP PEM', 'PRIVATE PUBLIC', 'alias') ¹	Lists key aliases in the MFT keystore. Example: listCSFKeyAliases('SSH', 'PRIVATE')
updateCSFKey	updKey	updateCSFKey('SSH PGP PEM', 'PRIVATE PUBLIC', 'alias', 'key_file_path') ¹	Deletes a key alias from the MFT keystore and generates a new key file. Example: updateCSFKey('SSH', 'PRIVATE', 'my-alias', '/export/ssh/my-private-key.ppk')
getSourceDeploymentHistory	getSrcDH	getSourceDeploymentHistory('source_name')	Returns the deployment history of a source artifact.
getTargetDeploymentHistory	getTrgtDH	getTargetDeploymentHistory('target_name')	Returns the deployment history of a target artifact.
getTransferDeploymentHistory	getXfrDH	getTransferDeploymentHistory('transfer_name')	Returns the deployment history of a transfer artifact.
getTransferInfo	getXfrInfo	getTransferInfo('transfer_name', 'label')	Returns information about a transfer artifact. Use Show Deployment Details on the Deployment tab to view the <i>label</i> .

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
pauseTransfer	pauseXfr	<code>pauseTransfer('instance_id', 'comment')</code>	Pauses an in-progress transfer. Open the Advanced section of the target report to view the instance ID. For information about the target report, see Interpreting Source, Transfer, and Target Reports . A comment is optional.
resubmit	resub	<code>resubmit('resubmit_type', 'instance_id', 'comment', 'IsSync')</code>	Resubmits a transfer. The <i>resubmit_type</i> is the type of artifact for which resubmit is invoked. Possible values are SOURCE, TRANSFER_INSTANCE, TARGET, or TARGET_INSTANCE. Open the Advanced section of the target report to view the instance ID. <i>IsSync</i> is a Boolean value for synchronous execution of resubmit. It is optional and the default value is FALSE. For information about the target report, see Interpreting Source, Transfer, and Target Reports . A comment is optional. Example: <pre>wls:/mydomain/ serverConfig> resubmit('SOURCE','3D48B1 2B-295A-4F52-A8EE- BD1CC1A20246', 'comments_for_resubmit', FALSE)</pre>

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
resubmitMessages	resMsgs	resubmitMessages(<i>resubmit_type</i> , <i>state</i> , <i>artifact_name</i> , <i>start_date</i> , <i>end_date</i> , <i>chunk_size</i> , <i>chunk_delay</i> , <i>ignore_ids</i> , <i>comments</i> , <i>preview_mode</i>)	Resubmits transfers in bulk. The <i>resubmit_type</i> is the type of artifact for which resubmit is invoked. Possible values are SOURCE, TRANSFER_INSTANCE, TARGET, or TARGET_INSTANCE. The <i>state</i> (optional) can be ACTIVE, FAILED, or COMPLETED. The <i>start_date</i> and <i>end_date</i> enable you to resubmit all failed messages within a specified date range in format dd-MM-yyyy H:m:s:S. When the command is run in preview mode (default=true), it lists the count of messages that will be resubmitted for given criteria. For more information, see Bulk Resubmit .
resumeTransfer	resXfer	resumeTransfer('instance_id', 'comment')	Resumes a paused transfer. Open the Advanced section of the target report to view the instance ID. For information about the target report, see Interpreting Source, Transfer, and Target Reports . A comment is optional.
configureHomeDir	confHmDir	configureHomeDir('directory_path', 'user_name')	Assigns the specified directory to the user as home directory where that user is located on login to embedded servers.
grantPermissionTo Directory	grPermDir	grantPermissionToDirectory('directory_path', 'principal_name', 'principal_type', 'permissions', 'server_type', 'include_subfolder')	Grants permission to an embedded server directory. Users and groups can be assigned a set of permissions to an existing directory on an embedded server.
listAllPermissions	lsPerms	listAllPermissions(<i>principal_name</i> , <i>server_types</i>)	Lists all permissions available for a given principal and server type. The server type can be FTP or sFTP. For example: wls:/mydomain/ serverConfig> listAllPermissions("weblogic", "FTP")
createArtifacts	crtAF	createArtifacts('xml_file_path', previewMode, updateIfExists)	Creates Artifacts from an input xml file containing artifact definition.

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
revokePermissionForDirectory	revPermDir	revokePermissionForDirectory('directory_path', 'principal_name', 'principal_type', 'permissions', 'server_type', 'include_subfolder')	Revokes a set of permissions from an embedded server directory.
startEmbeddedServer	startES	startEmbeddedServer('FTP FTPS SFTP')	Starts an embedded FTP, FTPS (FTP over SSL), or sFTP (SSH-FTP) server that was stopped.
stopEmbeddedServer	stopES	stopEmbeddedServer('FTP FTPS SFTP')	Stops an embedded FTP, FTPS (FTP over SSL), or sFTP (SSH-FTP) server that is running.
updatePorts	updPorts	updatePorts('server_instance_name', 'FTP FTPS SFTP', 'port')	Updates the port for an embedded FTP, FTPS (FTP over SSL), or sFTP (SSH-FTP) server, which is a service of an Oracle WebLogic Server managed server dedicated to MFT.
createCallouts	crtCalls	createCallouts('def_file_path')	Creates callouts based on an XML file that defines them.
deleteCallout	delCalls	deleteCallout('callout_name')	Deletes a callout.
listCallouts	lsCalls	listCallouts()	Lists callouts.
updateCallouts	updCalls	updateCallouts('def_file_path')	Updates callouts with the same names based on an XML file that defines them.
addContactToNotification	addContNotif	addContactToNotification('event', 'Email PHONE FAX SMS', 'value')	<p>Adds a contact to a specific event notification. The <i>value</i> is an email address or phone number.</p> <p>The <i>event</i> is RUNTIME_ERROR_EVENT, DELETE_ARTIFACT_EVENT, DEPLOY_ARTIFACT_EVENT, EXPORT_IMPORT_EVENT, PURGE_EVENT, or ARCHIVE_RESTORE_EVENT.</p>
createContact	crtCont	createContact('Email PHONE FAX SMS', 'value')	Creates a contact for event notifications. The <i>value</i> is an email address or phone number.
deleteContact	delCont	deleteContact('Email PHONE FAX SMS', 'value')	Deletes a contact. The <i>value</i> is an email address or phone number.
listContacts	lsConts	listContacts('Email PHONE FAX SMS')	Lists contacts.

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
removeContactFromNotification	remContNo te	removeContactFromNotification('event', 'Email PHONE FAX SMS', 'value')	Removes a contact from a specific event notification. The <i>value</i> is an email address or phone number. The <i>event</i> is RUNTIME_ERROR_EVENT, DELETE_ARTIFACT_EVENT, DEPLOY_ARTIFACT_EVENT, EXPORT_IMPORT_EVENT, PURGE_EVENT, or ARCHIVE_RESTORE_EVENT.
updateEvent	updEvt	updateEvent('event', 'enabled')	Enables or disables a specific event notification. Set <i>enabled</i> to TRUE or FALSE. The <i>event</i> is RUNTIME_ERROR_EVENT, DELETE_ARTIFACT_EVENT, DEPLOY_ARTIFACT_EVENT, EXPORT_IMPORT_EVENT, PURGE_EVENT, or ARCHIVE_RESTORE_EVENT.
archiveInstanceData	arcData	archiveInstanceData(archiveFileName='filename', startDate='date', endDate='date', batchId='batchId', status='C F A *', testMode='TRUE FALSE', comments='text', runInSync='FALSE TRUE', fsArchiveFolderPath='path')	Archives runtime instances to a .dmp file. The <i>archiveFileName</i> is required. Dates are in dd-MM-yyyy H:m:s:S format. The <i>batchId</i> is an identifier in the output of a previous <i>archiveInstanceData</i> command. The <i>status</i> is completed (default), failed, active, or all. To archive runtime instances, set <i>testMode</i> =FALSE. Set <i>runInSync</i> =TRUE to run immediately and block execution of other WLST commands. The <i>fsArchiveFolderPath</i> is required if archiving the corresponding payloads.
restoreInstanceData	resData	restoreInstanceData(archiveFilePath='path', fileNamePrefix='prefix', fsFolderPath='path', runInSync='FALSE TRUE')	Restores previously archived runtime instances. The <i>archiveFilePath</i> is required. The <i>fileNamePrefix</i> (usually <i>batchId</i>) and <i>fsFolderPath</i> are required if restoring the corresponding payloads. Set <i>runInSync</i> =TRUE to run immediately and block execution of other WLST commands.

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
archivePayloads	arcPLs	archivePayloads(batchId='batchId', archivePath='path', runInSync='FALSE TRUE')	Archives payloads corresponding to runtime instances to <i>batchId_n.zip</i> files. The <i>batchId</i> is an identifier in the output of a previous <code>archiveInstanceData</code> command and is required. The <i>archivePath</i> to the payload archive directory is also required. Set <code>runInSync=TRUE</code> to run immediately and block execution of other WLST commands.
restorePayloadsBy Name	resPLbyN	restorePayloadsByName(fileName='filename', folderPath='path', runInSync='FALSE TRUE')	Restores previously archived payloads by file name. The <i>fileNames</i> argument (usually <i>batchId</i>) is required. The <i>folderPath</i> to the payload archive directory is also required. Set <code>runInSync=TRUE</code> to run immediately and block execution of other WLST commands.
restorePayloadsBy Prefix	resPLbyP	restorePayloadsByPrefix(fileNamePrefix='prefix', folderPath='path', runInSync='FALSE TRUE')	Restores previously archived payloads by file name prefix. The <i>fileNamePrefix</i> argument (usually <i>batchId</i>) is required. The <i>folderPath</i> to the payload archive directory is also required. Set <code>runInSync=TRUE</code> to run immediately and block execution of other WLST commands.

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
purgeInstanceData	prgData	purgeInstanceData(startDate='date', endDate='date', batchId='batchId', status='C F A *', testMode='TRUE FALSE', comments='text', runInSync='FALSE TRUE', runPayloadPurge='FALSE TRUE', transfer_names, names_delimiter)	Purges runtime instances. All arguments are optional. Dates are in dd-MM-yyyy H:m:s:S format. The batchId is an identifier in the output of a previous archiveInstanceData or purgeInstanceData command. The status is completed (default), failed, active, or all. To archive runtime instances, set testMode=FALSE. Set runInSync=TRUE to run immediately and block execution of other WLST commands. Set runPayloadPurge=TRUE to purge the corresponding payloads. Set purgeTransactionTimeout MBean from EM to override the default time-out limit for purge operation.
purgePayloads	prgPLs	purgePayloads(batchId='batchId', detailedAudit='TRUE FALSE', runInSync='FALSE TRUE')	Purges payloads corresponding to runtime instances. The batchId is an identifier in the output of a previous archiveInstanceData or purgeInstanceData command and is required. Set detailedAudit=FALSE to turn off auditing of purged files. Set runInSync=TRUE to run immediately and block execution of other WLST commands.
createUserContact	crtUCont	createUserContact(user_name, delivery_channel)	Create a new user contact, which can be used for event notification. . <i>delivery_channel</i> (optional): possible values are EMAIL or SMS. If not specified, it will use the user preferred delivery channel configured in the WebLogic user.
createUserGroupContact	crtUGCont	createUserGroupContact(user_group_name, delivery_channel)	Create a new user group contact, which can be used for event notification. <i>delivery_channel</i> (optional): possible values are EMAIL or SMS. If not specified, it will use the user preferred delivery channel configured in the WebLogic user.

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
addUserContactToNotification	addUContN ote	addUserContactToNotification(<i>event</i> , <i>user_name</i> , <i>delivery_channel</i>)	<p>Add user contact for notification event.</p> <p>Event values: RUNTIME_ERROR_EVENT, DELETE_ARTIFACT_EVENT, DEPLOY_ARTIFACT_EVENT, EXPORT_IMPORT_EVENT</p> <p><i>delivery_channel</i> (optional): possible values are EMAIL or SMS.</p> <p>Note: Before adding an internal contact make sure that the email address/contact number is present in the user setting.</p>
addUserGroupContactToNotification	addUGCont Note	addUserGroupContactToNotification(<i>event</i> , <i>user_name</i> , <i>delivery_channel</i>)	<p>Add user group contact for notification event.</p> <p>Event values: RUNTIME_ERROR_EVENT, DELETE_ARTIFACT_EVENT, DEPLOY_ARTIFACT_EVENT, EXPORT_IMPORT_EVENT</p> <p><i>delivery_channel</i> (optional): possible values are EMAIL or SMS.</p>
deleteUserContact	delUCont	deleteUserContact(<i>user_name</i> , <i>delivery_channel</i>)	<p>Delete an existing user contact.</p> <p>Note: If the contact is in use (assigned to an event), the user will get an error message.</p> <p><i>delivery_channel</i> (optional): possible values are EMAIL or SMS.</p>
deleteUserGroupContact	delUGCont	deleteUserGroupContact(<i>user_group_name</i> , <i>delivery_channel</i>)	<p>Delete an existing user group contact. Note: If the contact is in use (assigned to an event), the user will get an error message.</p> <p><i>delivery_channel</i> (optional): possible values are EMAIL or SMS.</p>
removeUserContactFromNotification	remUContN ote	removeUserContactFromNotification(<i>event</i> , <i>user_name</i> , <i>delivery_channel</i>)	<p>Remove the given user contact from the notification event.</p> <p>Event values: RUNTIME_ERROR_EVENT, DELETE_ARTIFACT_EVENT, DEPLOY_ARTIFACT_EVENT, EXPORT_IMPORT_EVENT</p> <p><i>delivery_channel</i> (optional): possible values are EMAIL or SMS.</p>

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
removeUserGroupContactfromNotification	remUGCont Note	removeUserGroupContactFromNotification(<i>event</i> , <i>user_group_name</i> , <i>delivery_channel</i>)	Remove the given user group contact from the notification event. Event values: RUNTIME_ERROR_EVENT, DELETE_ARTIFACT_EVENT, DEPLOY_ARTIFACT_EVENT, EXPORT_IMPORT_EVENT <i>delivery_channel</i> (optional): possible values are EMAIL or SMS.
triggerEvent	trgEvt	triggerEvent('source_name', 'properties')	Trigger an Event on JCA/OCS/RIDC sources to initiate the file transfer. <i>source_name</i> : The name of the JCA/OCSS/RIDC source for which Event option is enabled. <i>properties</i> : Additional properties in the form of comma-separated <i>name=value</i> pairs.
updateTriggerEventStatus	updTrgEvt St	updateTriggerEventStatus(<i>status</i> , <i>source_name</i> , <i>event_session_id</i>)	Update the TriggerEvent status of the specified <i>event_session_id</i> . If no <i>event_session_id</i> is provided, then most recent event status of the given source is updated.
activatePurgeSchedule	actPurgeSch	activatePurgeSchedule(<i>schedule_name</i>)	Activate a purge schedule. Provide the purge schedule name to activate. Default <i>schedule_name</i> is the default purge schedule. If no purge schedule name is provided, the command will activate the default purge schedule. Only one purge schedule can be activated using this command.
deactivatePurgeSchedule	deactPurgeSch	deactivatePurgeSchedule(<i>schedule_name</i>)	Deactivate a purge schedule. Provide the purge schedule name to deactivate it. Default <i>schedule_name</i> is the default purge schedule. If no purge schedule name is provided, the command will deactivate the default purge schedule. Only one purge schedule can be deactivated using this command.

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
modifyPurgeSchedule	modifyPurgeSch	<code>modifyPurgeSchedule(start_date, end_date, schedule_time, frequency, retention_period, status, transfer_names, names_delimiter, include, comment)</code>	<p>Modify an existing purge schedule.</p> <p><i>start_date</i> is in <i>dd-mm-yyyy</i> format, <i>schedule_time</i> is in <i>hh:mm:ss</i> format, <i>frequency</i> values are DAILY, WEEKLY, MONTHLY, or YEARLY.</p> <p><i>retention_period</i> values are any non-negative number, <i>status</i> values are COMPLETED and/or FAILED, <i>transfer_names</i> can be Transfer names whose instances need to be purged, <i>names_delimiter</i> is a single character string, <i>include</i> values are TRUE or FALSE, <i>comment</i> value is any string.</p>

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
updateAppProperties	updAppPrt	<code>updateAppProperties('properties_name_value_pair', 'delimiter')</code>	Update Application properties of MFT. The supported properties are Server, HA, Performance and Advanced properties. Allows to update more than one property at a time where property is a name/value pair, name is the property name and value is the property value and each separated by a delimiter. Supported parameters are: physicalstoragedirectory: String - Directory Path, calloutdirectory: String - Directory Path, storeonlinepayload: String - {fileSystem, database}, storereferencepayload: Boolean - true/false, generatechecksum: Boolean - true/false, sourceprocessors: Number - non zero, non negative number, instanceprocessors: Number - non zero, non negative number, targetprocessors: Number - non zero, non negative number, controldirectory: String - Directory Path, inbounddatasource: String - MFT Data Source Name, outbounddatasource: String - MFT Data Source Name, internaladdress: String - IP Address of Load Balancer (LB), internalFTPS: Number - Port number of FTPS in LB, internalSFTP: Number - Port number of SFTP in LB, internalFTP: Number - Port number of FTP in LB, externaladdress: String - IP Address of LB, externalFTPS: Number - Port number of FTPS in LB, externalSFTP: Number - Port number of SFTP in LB, externalFTP: Number - Port number of FTP in LB.

Table 9-1 (Cont.) MFT WLST Command Summary

Command	Shortcut	Syntax	Description
updateSFTPServer	updSFTPSv	<code>updateSFTPServer('enable_ SFTP', 'key_alias', private_key_password')</code>	Used to enable/disable the Embedded SFTP Server. To enable SFTP server, SSH key alias name is mandatory along with the optional private key password. If the key is not password protected, then private key password is not required. To disable, both key alias and password are not required. Supported parameters are: <i>enable_SFTP</i> : Boolean to enable/disable SFTP, <i>key_alias</i> : SSH private key alias, <i>private_key_password</i> : Optional password of the SSH private key.

¹ An RSA key of PEM format is used to connect to Oracle Cloud Infrastructure when the OCI Storage Cloud Service type is selected as a source or target.

For an RSA key of PEM format, only PRIVATE is valid.

Oracle Managed File Transfer EJBs

WLST commands are also exposed as Enterprise Java Beans (EJB) and the commands are available in one of the following EJBs:

- `oracle.tip.mft.j2ee.ejb.KeyManagerService`
- `oracle.tip.mft.j2ee.ejb.MDSService`
- `oracle.tip.mft.j2ee.ejb.RuntimeService`