# Oracle® Application Server Integration InterConnect

Adapter for FTP Installation and User's Guide

10*g* Release 2 (10.1.2)

**Part No. B14073-01**

November 2004

Oracle Application Server Integration InterConnect Adapter for FTP Installation and User's Guide, 10*g* Release 2 (10.1.2)

Part No. B14073-01

Primary Author: Pradeep Vasudev, Vimmy K Raj

Contributing Author: Sivaraj Subbaiyan, Maneesh Joshi, Rahul Pathak, Harish Sriramulu

# Contents

# Send Us Your Comments

**Oracle Application Server Integration InterConnect Adapter for FTP Installation and User's Guide, 10*g* Release 2 (10.1.2)**

**Part No. B14073-01**

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appserverdocs_us@oracle.com
- FAX: 650-506-7375   Attn: Oracle Application Server Documentation Manager
- Postal service:

  Oracle Corporation  
  Oracle Application Server Documentation Manager  
  500 Oracle Parkway, M/S 1op6  
  Redwood Shores, CA  94065  
  USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

# Preface

This Preface contains these topics:

- Audience
- Documentation Accessibility
- Structure
- Related Documents
- Conventions

## Audience

*Oracle Application Server Integration InterConnect Adapter for FTP Installation and User's Guide* is intended for system administrators of OracleAS Integration InterConnect who perform the following tasks:

- install applications
- maintain applications

To use this document, you need to know how to install and configure OracleAS Integration InterConnect.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

### Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

## Structure

This document contains:

### Chapter 1, "Introduction"

This chapter describes the OracleAS Integration InterConnect Adapter for FTP (FTP adapter), and the hardware and software requirements.

### Chapter 2, "Installation and Configuration"

This chapter describes installation and configuration of the FTP adapter.

### Chapter 3, "Design Time and Runtime Concepts"

This chapter describes the design time and runtime concepts of the FTP adapter.

### Appendix A, "Frequently Asked Questions"

This chapter provides answers to frequently asked questions about the FTP adapter.

### Appendix B, "Example of the adapter.ini File"

This appendix shows an example of the `adapter.ini` file.

## Related Documents

For more information, refer to these Oracle resources:

- Oracle Application Server Integration InterConnect User's Guide
- Oracle Application Server Integration InterConnect Installation Guide

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://www.oracle.com/technology/membership/`

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://www.oracle.com/technology/documentation/`

## Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text

- Conventions in Code Examples
- Conventions for Windows Operating Systems

## Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

| Convention | Meaning | Example |
|---|---|---|
| **Bold** | Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both. | When you specify this clause, you create an **index-organized table**. |
| *Italics* | Italic typeface indicates book titles or emphasis. | *Oracle Database 10g Concepts*<br>Ensure that the recovery catalog and target database do *not* reside on the same disk. |
| `UPPERCASE monospace (fixed-width) font` | Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, Recovery Manager keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, user names, and roles. | You can specify this clause only for a `NUMBER` column.<br>You can back up the database by using the `BACKUP` command.<br>Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view.<br>Use the `DBMS_STATS.GENERATE_STATS` procedure. |
| `lowercase monospace (fixed-width) font` | Lowercase monospace typeface indicates executable programs, filenames, directory names, and sample user-supplied elements.<br>*Note:* Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown. | Enter `sqlplus` to start SQL*Plus.<br>The password is specified in the `orapwd` file.<br>Back up the datafiles and control files in the `/disk1/oracle/dbs` directory.<br>The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table.<br>Connect as `oe` user.<br>The `JRepUtil` class implements these methods. |
| *`lowercase italic monospace (fixed-width) font`* | Lowercase italic monospace font represents placeholders or variables. | You can specify the *`parallel_clause`*.<br>Run *`old_release`*`.SQL` where *`old_release`* refers to the release you installed prior to upgrading. |

## Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

| Convention | Meaning | Example |
|---|---|---|
| `[ ]` | Anything enclosed in brackets is optional. | `DECIMAL (`*`digits`*` [ , `*`precision`*` ])` |
| `{ }` | Braces are used for grouping items. | `{ENABLE \| DISABLE}` |

| Convention | Meaning | Example |
|---|---|---|
| \| | A vertical bar represents a choice of two options. | `{ENABLE \| DISABLE}`<br>`[COMPRESS \| NOCOMPRESS]` |
| ... | Ellipsis points mean repetition in syntax descriptions.<br>In addition, ellipsis points can mean an omission in code examples or text. | CREATE TABLE ... AS *subquery*;<br>SELECT *col1*, *col2*, ... , *coln* FROM employees; |
| Other symbols | You must use symbols other than brackets ([ ]), braces ({ }), vertical bars (\|), and ellipsis points (...) exactly as shown. | `acctbal NUMBER(11,2);`<br>`acct    CONSTANT NUMBER(4) := 3;` |
| *Italics* | Italicized text indicates placeholders or variables for which you must supply particular values. | CONNECT SYSTEM/*system_password*<br>DB_NAME = *database_name* |
| UPPERCASE | Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. Because these terms are not case sensitive, you can use them in either UPPERCASE or lowercase. | SELECT last_name, employee_id FROM employees;<br>SELECT * FROM USER_TABLES;<br>DROP TABLE hr.employees; |
| lowercase | Lowercase typeface indicates user-defined programmatic elements, such as names of tables, columns, or files.<br>**Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown. | SELECT last_name, employee_id FROM employees;<br>sqlplus hr/hr<br>CREATE USER mjones IDENTIFIED BY ty3MU9; |

### Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

| Convention | Meaning | Example |
|---|---|---|
| Click **Start**, and then choose the ***menu item*** | How to start a program. | To start the Database Configuration Assistant, click **Start**, and choose **Programs**. In the Programs menu, choose **Oracle** - ***HOME_NAME*** and then click **Configuration and Migration Tools**. Choose **Database Configuration Assistant**. |
| File and directory names | File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (\|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the filename begins with \\, then Windows assumes it uses the Universal Naming Convention. | c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32 |

| Convention | Meaning | Example |
|---|---|---|
| `C:\>` | Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual. | `C:\oracle\oradata>` |
| Special characters | The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters. | `C:\>exp HR/HR TABLES=employees QUERY=\"WHERE job_id='SA_REP' and salary<8000\"` |
| *HOME_NAME* | Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore. | `C:\> net start Oracle`*HOME_NAME*`TNSListener` |
| *ORACLE_HOME* and *ORACLE_BASE* | In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory.<br>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle\product\10.1.0`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\product\10.1.0\db_`*n*, where *n* is the latest Oracle home number. The Oracle home directory is located directly under *ORACLE_BASE*.<br>All directory path examples in this guide follow OFA conventions.<br>Refer to *Oracle10g Database Installation Guide for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories. | Chnage to the *ORACLE_BASE*\*ORACLE_HOME*`\rdbms\admin` directory. |

# 1 Introduction

This chapter provides an overview on how to use Oracle Application Server Integration InterConnect (OracleAS Integration InterConnect) Adapter for File Transfer Protocol (FTP adapter). It contains the following:

## 1.1 FTP Adapter Overview

The FTP adapter enables an FTP application to be integrated with other applications using OracleAS Integration InterConnect. The FTP adapter is useful in all Enterprise Application Integration (EAI) environments that use FTP. EAI is the integration of applications and business processes within the same company.

The FTP adapter can monitor incoming messages which are in the form of FTP files placed in a remote FTP server or on local file systems. The FTP adapter is also capable of sending messages to remote FTP servers by proxy host. The payload for this adapter can be XML data or D3L data.

> **See Also:** *Oracle Application Server Integration InterConnect User's Guide*

Figure 1–1 shows the data flow of incoming messages, from remote FTP server/local file system to OracleAS Integration InterConnect.

*Figure 1–1 Incoming Messages Diagram*

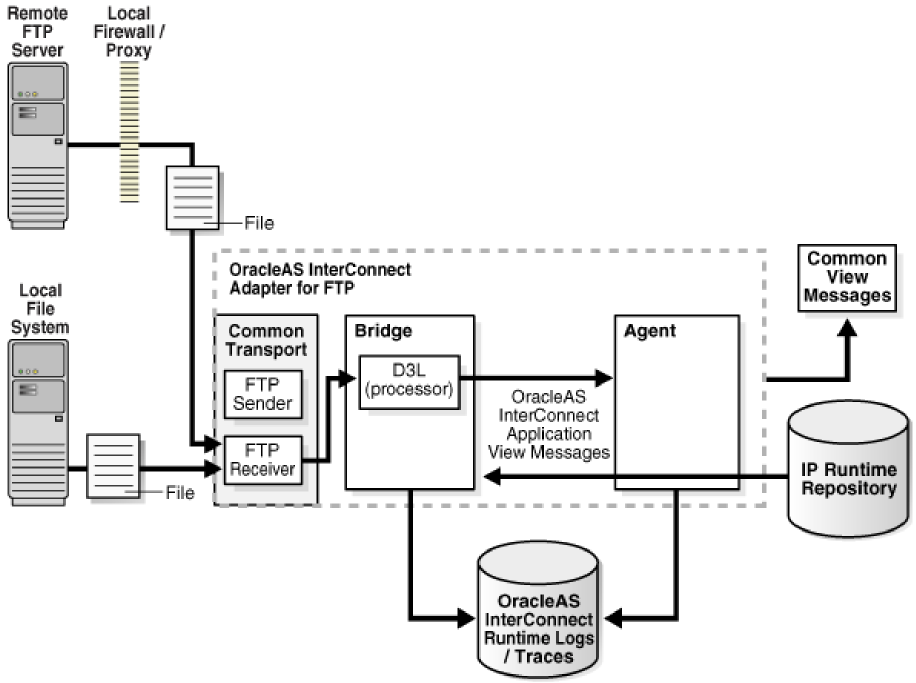
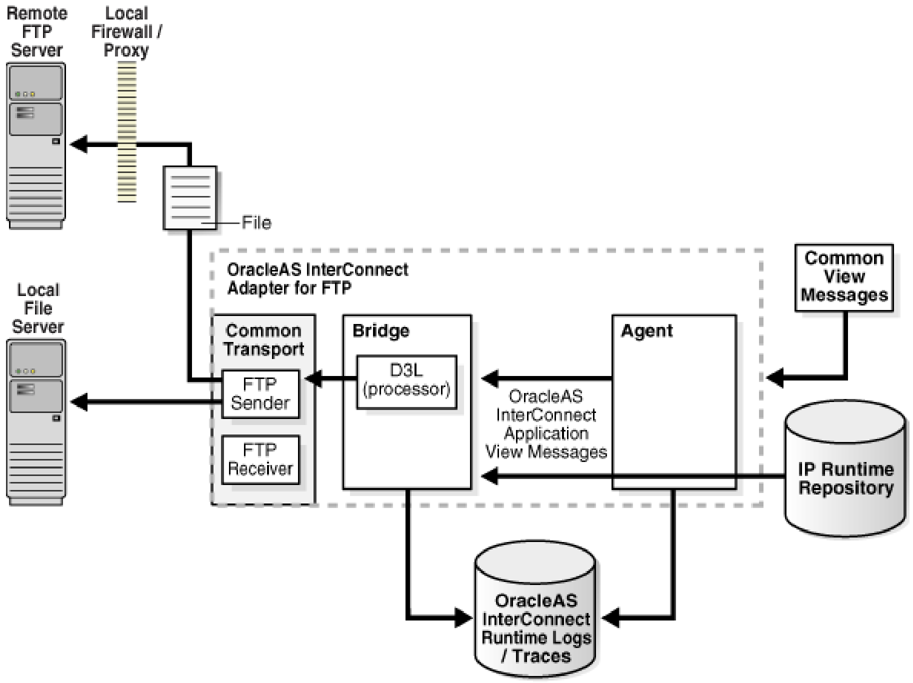


Figure 1–2 shows the data flow of outgoing messages from OracleAS Integration InterConnect to the remote FTP Server.

*Figure 1–2 Outgoing Messages Diagram*



## 1.2 FTP Adapter System Requirements

The following sections describe the FTP adapter system requirements:

- Hardware Requirements
- Software Requirements

### 1.2.1 Hardware Requirements

Table 1–1 lists the hardware requirements for the computer where the FTP adapter will be installed.

***Table 1–1 Hardware Requirements***

| Hardware | Windows | UNIX |
|---|---|---|
| Disk Space | 500 MB | 500 MB |
| Memory | 128 MB | 128 MB |

### 1.2.2 Software Requirements

The following sections describe the FTP adapter software requirements:

- Operating System Requirements
- JRE Requirements

#### Operating System Requirements

Table 1–2 lists the operating system requirements for the computer where the FTP adapter will be installed.

***Table 1–2 Operating System Requirements***

| Operating System | Version |
|---|---|
| HP Tru64 | HP Tru64 UNIX (Alpha) 5.1b |
| HP-UX | HP-UX (PA-RISC) 11.11, 11.23 |
| IBM AIX | AIX (POWER) version 5.2 |
| Linux (x86) | Red Hat Enterprise Linux 2.1, 3.0<br>SuSE SLES8, SLES9 |
| Sun SPARC Solaris | Sun SPARC Solaris 2.8 and 2.9 |
| Microsoft Windows | Windows XP Professional, Windows 2000( SP3 or higher) |

#### JRE Requirements

OracleAS Integration InterConnect uses Java Runtime Environment (JRE) 1.4, which is installed with its components.

## 1.3 Known FTP Adapter Limitations

The FTP adapter has the following limitations:

- Does not support secure FTP.
- Does not support file-filtering features.
- Can only receive from a single FTP or file endpoint.

- The sending endpoint and receiving endpoint are restricted to FTP and file endpoints.

# 2 Installation and Configuration

This chapter describes how to install and configure the FTP adapter. It contains the following topics:

- Installing the FTP Adapter
- Configuring the FTP Adapter

## 2.1 Installing the FTP Adapter

The FTP adapter must be installed in an existing Oracle home Middle Tier for Oracle10*g* InterConnect 10*g* Release 2 (10.1.2).

This section contains the following topics:

- Preinstallation Tasks
- Installation Tasks
- Postinstallation Tasks

### 2.1.1 Preinstallation Tasks

Consult the following guides before installing the FTP adapter:

- *Oracle Application Server Installation Guide*for information about Oracle Universal Installer startup.
- *Oracle Application Server Integration InterConnect Installation Guide* for information on mounting CD-ROMs, software, hardware, and system requirements for OracleAS Integration InterConnect.

> **Note:** OracleAS Integration InterConnect Hub is installable through the OracleAS Integration InterConnect Hub installation type. You must install the OracleAS Integration InterConnect Hub before proceeding with the FTP adapter installation.

### 2.1.2 Installation Tasks

To install the FTP adapter:

1. In the Available Product Components page of the OracleAS Integration InterConnect installation, select **FTP adapter**, and click **Next**.
2. The Set Oracle Wallet Password screen is displayed. Enter and confirm the password, which will be used to administer OracleAS Integration InterConnect installation. Click **Next**.

- Go to step 3, if installing the FTP adapter in an OracleAS Middle Tier Oracle home that does not have an InterConnect component already installed. Ensure that the OracleAS Integration InterConnect hub has been installed.
- Go to step 4, if installing the FTP adapter in an OracleAS Middle Tier Oracle home that has an existing InterConnect component. Ensure that it is a home directory to an OracleAS Integration InterConnect component.

**Note:** All passwords are stored in Oracle Wallet. Refer to How do I secure my passwords? for more details on how to modify and retrieve the password using Oracle Wallet.

3. The Specify Hub Database Connection screen is displayed. Enter information in the following fields:
   - Host Name: The host name of the computer where the hub database is installed.
   - Port Number: The TNS listener port for the hub database.
   - Database SID: The System Identifier (SID) for the hub database.
   - Password: The password for the hub database user.
4. Click **Next**. The Specify FTP adapter Name page is displayed.
5. Enter the application to be defined. Blank spaces are not permitted. The default value is `myFTPApp`.
6. Click **Next**. The Specify FTP adapter Usage screen is displayed.
7. Select one of the following options and go to the step specified.

| If You Select... | Then Click Next and Go to Step... |
|---|---|
| Configure for both sending and receiving messages | 8 |
| Configure for sending messages ONLY | 8 |
| Configure for receiving messages ONLY | 10 |

**Note:** You can change the values for these selections later by editing the parameter settings in the `adapter.ini` file.

8. Enter the following information in the Configure Sending Endpoint Information page:
   - Username: The user name for the FTP server.
   - Password: The user password for the FTP server.
   - FTP Mode: The mode of access used to send information to the specified URL. Select either binary or ASCII.
   - URL: The URL to be used for sending information. Enter one of the following:
     * For sending to an FTP server: *`ftp://host name/path`*
     * For sending to a local file system: *`file://localhost/path`*

**Note:** If the sender endpoint is a local file system, then the user name, password, and file type are not required.

9. Click **Next**. The installation page that displays next is based on the selection you made in Step 7.

| If You Selected... | Then Go to Step... |
|---|---|
| Configure for both sending and receiving messages | 10 |
| Configure for sending messages ONLY | 11 |

10. Enter the following information in the Configure Receiving Endpoint Information page:
    - Username: The user name account of the FTP server from which the Oracle Application Server Integration InterConnect Hub receives messages.
    - Password: The password for the user name account.
    - FTP Mode: The mode of access used to receive information from the specified URL. Select either binary or ASCII.
    - URL: The FTP URL to be used for receiving information. Enter one of the following:
      * For sending to an FTP server: `ftp://host name/path`
      * For sending to a local file system: `file://localhost/path`

**Note:** If the sender endpoint is a local file system, then the user name, password, and file type are not required.

**Caution:** During runtime, the FTP adapter connects to the FTP server or accesses the file system and removes the files in the directory specified by the receiving endpoint after processing. Oracle recommends you create a dedicated FTP account or user account (if the local file system is used for the receiving endpoint) for testing and deploying this adapter.

11. Click **Next**. The Summary page is displayed.
12. Click **Install** to install the FTP adapter. The adapter is installed in the following directory, depending on the operating system:

| Platform | Directory |
|---|---|
| UNIX | *ORACLE_HOME*/integration/interconnect/adapters/*Application* |
| Windows | *ORACLE_HOME*\integration\interconnect\adapters\*Application* |

You defined the value of `Application` in Step 5.

13. Click **Exit** on the End of Installation page to exit the FTP adapter installation.

## 2.1.3 Postinstallation Tasks

FTP adapter installation creates the `adapter.ini` file that consists of configuration parameters read by the FTP adapter at startup. These configuration parameter settings are suitable for most FTP application environments. To customize the `adapter.ini` file parameter settings for the FTP application, refer to the following sections:

- Customizing the Payload Datatype
- Customizing the Sending Endpoints
- Customizing the Receiving Endpoints

### 2.1.3.1 Customizing the Payload Datatype

Payload data is the data sent between applications. To change the payload datatype from the default of XML to the Data Definition Description Language (D3L), edit the following parameters in the `adapter.ini` file.

1. Set the `ota.type` parameter to the payload type D3L.

```
ota.type=D3L
```

2. Copy the D3L XML files associated with the FTP application to the directory in which the `adapter.ini` file is located.

3. Set the `ota.d3ls` parameter to specify the D3L files associated with the FTP application.

```
ota.d3ls=person1.xml,person2.xml
```

> **See Also:** `ota.type` and `ota.d3ls` parameter descriptions in Table 2–9

### 2.1.3.2 Customizing the Sending Endpoints

To customize the behavior of the sending endpoints (destinations) for messages, edit the following parameters in the `adapter.ini` file.

1. Edit the `ota.send.endpoint` parameter or leave it blank if it acts only as a receiver. For example:

   - For a remote file system. `ota.send.endpoint=ftp://foo.com/test`
   - For a local file system. `ota.send.endpoint=file://localhost/test`

   If the endpoint is a local file system, then leave the following parameters blank:

   - `file.sender.file_type`
   - `file.sender.password`
   - `file.sender.proxy_host`
   - `file.sender.proxy_port`

2. Set the `file.sender.file_type` parameter to the file type used in FTP. For example:

```
file.sender.file_type=BINARY
```

3. Update the `file.sender.user` and `file.sender.password` parameters with the information of the FTP account that serves as the sending endpoint.

4. If a proxy host is needed, then enter the values for the `file.sender.proxy_host` and `file.sender.proxy_port` parameters.
5. Set the `file.sender.staging_dir` parameter. This parameter prevents partial files picked up by external applications.
6. Set the `file.sender.file_name_rule` parameter. This parameter controls how the adapter generates the file name.
7. If you need to modify the contents of an outgoing message before it is sent by the transport layer, then you can customize it by implementing the `FileSenderCustomizer` interface. You need to set the `file.sender.customize_class` to the name of the customizing class.

   **See Also:** Section 3.2.2, "FTP Sender"

#### 2.1.3.3 Customizing the Receiving Endpoints

To customize the behavior of the receiving FTP or file endpoints for messages, edit the following parameters in the `adapter.ini` file.

1. Edit the `ota.receive.endpoint` parameter or leave it blank if the adapter only acts as a sender. For example:
   - For a remote file system: `ota.receive.endpoint=ftp://foo.com/test`
   - For a local file system: `ota.receive.endpoint=ftp://localhost/test`

   If the endpoint is a local file system, then leave the following parameter blank:
   - `file.receiver.file_type`
   - `file.receiver.password`
   - `file.receiver.proxy_host`
   - `file.receiver.proxy_port`

   **Warning: Do not set the ota.receive.endpoint parameter to a personal file directory as the files in that directory will be consumed and deleted by the FTP adapter after processing.**

2. Set the `file.receiver.file_type` parameter to the file type used in FTP. For example:

   ```
   file.receiver.file_type=BINARY
   ```

3. Update the `file.receiver.user` and `file.receiver.password` parameters with the information of the FTP account that serves as the receiving endpoint.
4. If a proxy host is needed, then enter the required values for the `file.receiver.proxy_host` and `file.receiver.proxy_port` parameters.
5. Set the `file.exception.exception_dir` to a local file system directory that stores files. For example:

   ```
   file.receiver.exception_dir=/tmp/error
   ```

6. Set the `file.receiver.polling_interval` parameter to the time interval in milliseconds to poll the FTP server or local file system. For example:

   ```
   file.receiver.polling_interval=2000
   ```

7. Set the `file.receiver.max_msgs_retrieved` parameter to the maximum number of messages to retrieve in polling a session. For example:

   ```
   file.receiver.max_msgs_retrieved=10
   ```

8. If you need to modify the contents of an incoming message before it is processed by the bridge, such as, to remove an extra line in a file, then you customize it by implementing the `FileSenderCustomizer` interface. You need to set the `file.receiver.customize_class` to the name of the customizing class.

## 2.2 Configuring the FTP Adapter

After an FTP adapter installation, you can configure it for your needs.The following tables describe the location and details of the configuration files.

Table 2–1 describes the location where the adapter is installed.

***Table 2–1 FTP Adapter Directory***

| Platform | Directory |
|---|---|
| UNIX | *ORACLE_HOME*/integration/interconnect/adapters/*Application* |
| Windows | *ORACLE_HOME*\integration\interconnect\adapters\*Application* |

Table 2–2 describes the executable files of the FTP adapter.

***Table 2–2 FTP Executable Files***

| File | Description |
|---|---|
| `start` (UNIX) | Does not use parameters, starts the adapter. |
| `start.bat` (Windows) | Does not use parameters, starts the adapter. |
| `stop` (UNIX) | Does not use parameters, stops the adapter. |
| `stop.bat` (Windows) | Does not use parameters, stops the adapter. |

Table 2–3 describes the FTP adapter configuration files.

***Table 2–3 FTP Configuration Files***

| File | Description |
|---|---|
| `adapter.ini` (UNIX) | Consists of all the initialization parameters that the adapter reads at startup. |
| `adapter.ini` (Windows) | Consists of all the initialization parameters that the adapter reads at startup. |

Table 2–4 describes the directories used by the FTP adapter.

***Table 2–4 FTP Directories***

| Directory | Description |
|---|---|
| `logs` | The adapter activity is logged in subdirectories of the `logs` directory. Each time the adapter is run, a new subdirectory is created for the `oailog.txt` log file. |

*Table 2–4 (Cont.) FTP Directories*

| Directory | Description |
|---|---|
| persistence | The messages are made available in this directory. Do not edit this directory or its files. |

## 2.2.1 Using the Application Parameter

Adapters do not have integration logic. The FTP adapter has a generic transformation engine that uses metadata from the repository as runtime instructions to perform transformations. The application parameter defines the capabilities of an adapter, such as the messages to be published and subscribed, and the transformations to be performed. The application parameter allows the adapter to retrieve only the relevant metadata from the repository. The application parameter must match the corresponding application name that will be defined in iStudio under the Applications folder.

If you use prepackaged metadata, then import it into the repository and start iStudio to find the corresponding application under the Applications folder. You can use this as the application name for the adapter you are installing.

**See Also:** Step 4 on page 2-2

## 2.2.2 Ini File Settings

The following are the `.ini` files used to configure the FTP adapter:

- hub.ini Files
- adapter.ini Files

### 2.2.2.1 hub.ini Files

The FTP adapter connects to the hub database using parameters in the `hub.ini` file located in the `hub` directory. Table 2–5 lists the parameter names, descriptions for each parameter, and examples.

*Table 2–5 hub.ini Parameters*

| Parameter | Description | Example |
|---|---|---|
| hub_host | The name of the computer hosting the hub database. There is no default value. The value is set during installation. | `hub_host=mpscottpc` |
| hub_instance | The SID of the hub database. There is no default value. The value is set during installation. | `hub_instance=orcl` |
| hub_port | The TNS listener port number for the hub database instance. There is no default value. The value is set during installation. | `hub_port=1521` |
| hub_username | The name of the hub database schema (or user name). The default value is `ichub`. | `hub_username=ichub` |
| repository_name | The name of the repository that communicates with the adapter. The default value is `InterConnectRepository`. | `repository_name=InterConnectRepository` |

#### Oracle Real Application Clusters hub.ini Parameters

When a hub is installed on an Oracle Real Application Clusters database, the parameters listed in Table 2–6 represent information about additional nodes used for connection and configuration. These parameters are in addition to the default parameters for the primary node. In Table 2–6, x represents the node number. The number is between 2 and the number of nodes. For example, if the cluster setup contains 4 nodes, x can be a value between 2 and 4.

*Table 2–6 Oracle Real Application Clusters hub.ini Parameters*

| Parameter | Description | Example |
|---|---|---|
| hub_hostx | The host where the Real Application Clusters database is installed. | `hub_host2=dscott13` |
| hub_instancex | The instance on the respective node. | `hub_instance2=orcl2` |
| hub_num_nodes | The number of nodes in a cluster. | `hub_num_nodes=4` |
| hub_portx | The port where the TNS listener is listening. | `hub_port2=1521` |

#### 2.2.2.2 adapter.ini Files

The agent component of the FTP adapter reads the `adapter.ini` file at runtime to access FTP adapter parameter configuration information. Table 2–7 lists the parameter names, descriptions for each parameter, and examples.

*Table 2–7 adapter.ini Parameters*

| Parameter | Description | Example |
|---|---|---|
| agent_admin_port | Specifies the port through which the adapter can be accessed through firewalls.<br>Possible Value: A valid port number.<br>Default Value: None. | `agent_admin_port=1059` |
| agent_delete_file_cache_at_startup | Specifies whether to delete the cached metadata during startup. If any agent caching method is enabled, then metadata from the repository is cached locally on the file system. Set the parameter to `true` to delete all cached metadata on startup.<br>Possible Values: `true` or `false`.<br>Default Value: `false`.<br>**Note:** After changing metadata or DVM tables for the adapter in iStudio, you must delete the cache to guarantee access to new metadata or table information. | `agent_delete_file_cache_at_startup=false` |
| agent_dvm_table_caching | Specifies the Domain Value Mapping (DVM) table caching algorithm.<br>Possible values:<br>■ `startup`: Cache all DVM tables at startup. This may be time-consuming if there are many tables in the repository.<br>■ `demand`: Cache tables as they are used.<br>■ `none`: No caching. This slows down performance.<br>Default Value: `demand`. | `agent_dvm_table_caching=demand` |

*Table 2–7 (Cont.) adapter.ini Parameters*

| Parameter | Description | Example |
|---|---|---|
| agent_log_level | Specifies the amount of logging necessary.<br>Possible values:<br>`0`=errors only<br>`1`=status and errors<br>`2`=trace, status, and errors<br>Default Value: 1. | `agent_log_level=2` |
| agent_lookup_table_caching | Specifies the lookup table caching algorithm.<br>Possible values:<br>- `startup`: Cache all lookup tables at startup. This may be time-consuming if there are many tables in the repository.<br>- `demand`: Cache tables as they are used.<br>- `none`: No caching. This slows down performance.<br>Default Value: `demand`. | `agent_lookup_table_caching=demand` |
| agent_max_ao_cache_size | Specifies the maximum number of application object metadata to cache.<br>Possible Value: An integer greater than or equal to `1`.<br>Default Value: `200`. | `agent_max_ao_cache_size=200` |
| agent_max_co_cache_size | Specifies the maximum number of common object metadata to cache.<br>Possible Value: An integer greater than or equal to `1`.<br>Default Value: `100`. | `agent_max_co_cache_size=100` |
| agent_max_dvm_table_cache_size | Specifies the maximum number of DVM tables to cache.<br>Possible Value: An integer greater than or equal to `1`.<br>Default Value: `200`. | `agent_max_dvm_table_cache_size=200` |
| agent_max_lookup_table_cache_size | Specifies the maximum number of lookup tables to cache.<br>Possible Value: Any integer greater than or equal to `1`.<br>Default Value: `200`. | `agent_max_lookup_table_cache_size=200` |
| agent_max_message_metadata_cache_size | Specifies the maximum number of message metadata (publish/subscribe and invoke/implement) to cache.<br>Possible Value: An integer greater than or equal to `1`.<br>Default Value: `200`. | `agent_max_message_metadata_cache_size=200` |
| agent_max_queue_size | Specifies the maximum size internal OracleAS Integration InterConnect message queues can grow.<br>Possible Value: An integer greater than or equal to `1`.<br>Default Value: `1000`. | `agent_max_queue_size=1000` |
| agent_message_selector | Specifies conditions for message selection when the adapter registers its subscription with the hub.<br>Possible Value: A valid Oracle Advanced Queue message selector string (like `'%,aqapp,%'`).<br>Default Value: None. | `agent_message_selector=%,aqapp,%` |

*Table 2–7 (Cont.) adapter.ini Parameters*

| Parameter | Description | Example |
|---|---|---|
| agent_metadata_caching | Specifies the metadata caching algorithm.<br>Possible values:<br>- `startup`: Cache everything at startup. This may be time-consuming if there are many tables in the repository.<br>- `demand`: Cache metadata as it is used.<br>- `none`: No caching. This slows down performance.<br>Default Value: `demand`. | `agent_metadata_caching=demand` |
| agent_persistence_cleanup_interval | Specifies how often to run the persistence cleaner thread in milliseconds.<br>Possible Value: An integer greater than or equal to `30000` milliseconds.<br>Default Value: `60000`. | `agent_persistence_cleanup_interval=60000` |
| agent_persistence_queue_size | Specifies the maximum size of internal OracleAS Integration InterConnect persistence queues.<br>Possible Value: An integer greater than or equal to `1`.<br>Default Value: `1000`. | `agent_persistence_queue_size=1000` |
| agent_persistence_retry_interval | Specifies how often the persistence thread retries when it fails to send an OracleAS Integration InterConnect message.<br>Possible Value: An integer greater than or equal to `5000` milliseconds.<br>Default Value: `60000`. | `agent_persistence_retry_interval=60000` |
| agent_pipeline_from_hub | Specifies whether to turn on the pipeline for messages from the hub to the bridge. If you set the pipeline to `false`, then the file persistence is not used in that direction.<br>Possible Value: `true`, `false`<br>Default Value: `false`. | `agent_pipeline_from_hub=false` |
| agent_pipeline_to_hub | Specifies whether to turn on the pipeline for messages from the bridge to the hub. If you set the pipeline to `false`, then the file persistence is not used in that direction.<br>Possible Value: `true`, `false`.<br>Default Value: `false`. | `agent_pipeline_to_hub=false` |
| agent_reply_message_selector | Specifies the application instance to which the reply must be sent. This parameter is used if multiple adapter instances exist for the given application and given partition.<br>Possible Value: A string built using the application name (parameter:application) concatenated with the instance number (parameter:instance_number).<br>Default Value: None. | If `application=aqapp, instance_number=2,`then `agent_reply_message_selector=`recipient_list `like'%,aqapp2,%'` |

*Table 2–7 (Cont.) adapter.ini Parameters*

| Parameter | Description | Example |
|---|---|---|
| agent_reply_ subscriber_name | Specifies the subscriber name used when multiple adapter instances are used for the given application and given partition. This parameter is optional if only one instance is running.<br>Possible Value: The application name (parameter:application) concatenated with the instance number (parameter:instance_number).<br>Default Value: None. | If `application=ftpapp` and `instance_number=2`, then `agent_reply_ subscriber_ name=ftpapp2` |
| agent_subscriber_ name | Specifies the subscriber name used when this adapter registers its subscription.<br>Possible Value: A valid Oracle Advanced Queue subscriber name.<br>Default Value: None. | `agent_subscriber_ name=ftpapp` |
| agent_throughput_ measurement_ enabled | Specifies if the throughput measurement is enabled. Set this parameter to `true` to turn on throughput measurements.<br>Default Value: `true`. | `agent_throughput_ measurement_ enabled=true` |
| agent_tracking_ enabled | Specifies if message tracking is enabled. Set this parameter to `false` to turn off tracking of messages. Set this parameter to `true` to track messages with tracking fields set in iStudio.<br>Default Value: `true`. | `agent_tracking_ enabled=true` |
| agent_use_custom_ hub_dtd | Specifies whether to use a custom DTD for the common view message when handing it to the hub. By default, adapters use a specific OracleAS Integration InterConnect DTD for all messages sent to the hub.<br>Set this parameter to `true` to have the adapter use the DTD imported for the message of the common view instead of the OracleAS Integration InterConnect DTD.<br>Default Value: None. | `agent_use_custom_hub_ dtd=false` |
| application | Specifies the name of the application to which this adapter connects. This must match the name specified in iStudio while creating metadata.<br>Possible Value: An alphanumeric string.<br>Default Value: None. | `application=ftpapp` |
| encoding | Specifies the character encoding for published messages. The adapter uses this parameter to generate encoding information for the encoding tag of transformed OracleAS Integration InterConnect messages. OracleAS Integration InterConnect represents messages internally as XML documents.<br>Possible Value: A valid character encoding.<br>Default Value: `UTF-8`.<br>When there is no existing encoding in the subscribed message, this parameter will be used to explicitly specify the encoding of the published message. This parameter will be ignored when the encoding already exists in the subscribed message. | `encoding=Shift_JIS` |

*Table 2–7 (Cont.) adapter.ini Parameters*

| Parameter | Description | Example |
|---|---|---|
| external_dtd_base_url | Specify the base URL for loading external enitites and DTDs.This specifies to the XML parser to resolve the external entities in the instance document using the given URL.<br>Possible Value: A URL.<br>Default Value: The URL of the current user directory. | external_dtd_base_url=file://C:\InterConnect10_1_2\adapters\AQApp\ |
| instance_number | Specifies the instance number to which this adapter corresponds. Specify a value only if you have multiple adapter instances for the given application with the given partition.<br>Possible Value: An integer greater than or equal to `1`.<br>Default Value: None. | `instance_number=1` |
| nls_country | Specifies the ISO country code. The codes are defined by ISO-3166.<br>Possible Value: A valid code. A full list of the codes is available at `http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html`<br>Default Value: `US`.<br>**Note**: This parameter specifies date format and is applicable for the date format only. | `nls_country=US` |
| nls_date_format | Specifies the format for a date field expressed as a string.<br>Possible Value: A valid date format pattern as shown in Table 2–8 for the definitions of the format characters.<br>Default Value: `EEE MMM dd HHmmss zzz yyyy`. | Date format pattern `dd/MMM/yyyy` can represent 01/01/2003.<br>`nls_date_format=dd-MMM-yy`<br>Multiple date formats can be specified as `num_nls_formats=2`<br>`nls_date_format1=dd-MMM-yy`<br>`nls_date_format2=dd/MMM/yy` |
| nls_language | Specifies the ISO language code. The codes are defined by ISO-639.<br>Possible Value: A valid code. A full list of these codes is available at `http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt`<br>Default Value: `en`.<br>**Note**: This parameter specifies date format and is applicable for the date format only. | `nls_language=en` |
| partition | Specifies the partition this adapter handles as specified in iStudio.<br>Possible Value: An alphanumeric string.<br>Default Value: None. | `partition=germany` |

***Table 2–7 (Cont.) adapter.ini Parameters***

| Parameter | Description | Example |
|---|---|---|
| service_class | Specifies the entry class for the Windows service.<br>Possible Value: `oracle/oai/agent/service/AgentService`.<br>Default Value: None. | `service_class=oracle/oai/agent/service/AgentService` |
| service_classpath | Specifies the class path used by the adapter JVM. If a custom adapter is developed and the adapter is to pick up any additional jar files, then add the files to the existing set of jar files.<br>Possible Value: A valid `PATH` setting.<br>Default Value: None.<br>This parameter is for Microsoft Windows only. | `service_classpath=D:\oracle\oraic\integration\interconnect\lib\oai.jar;D:\oracle\oraic\jdbc\classes12.zip` |
| service_jdk_dll | Specifies the Dynamic Link Library(DLL) that the adapter JVM should use.<br>Possible Value: A valid `jvm.dll`.<br>Default Value: `jvm.dll`.<br>This parameter is for Microsoft Windows only. | `service_jdk_dll=jvm.dll` |
| service_jdk_version | Specifies the JDK version that the adapter JVM should use.<br>Possible Value: A valid JDK version number.<br>Default Value: 1.4<br>This parameter is for Microsoft Windows only. | `service_jdk_version=1.4` |
| service_max_heap_size | Specifies the maximum heap size for the adapter JVM.<br>Possible Value: A valid JVM heap size.<br>Default Value: `536870912`.<br>This parameter is for Microsoft Windows only. | `service_max_heap_size=536870912` |
| service_max_java_stack_size | Specifies the maximum size the JVM stack can grow.<br>Possible Value: A valid JVM maximum stack size.<br>Default Value: Default value for the JVM.<br>This parameter is for Microsoft Windows only. | `service_max_java_stack_size=409600` |
| service_max_native_stack_size | Specifies the maximum size the JVM native stack can grow.<br>Possible Value: A valid JVM maximum native stack size.<br>Default Value: Default value for the JVM.<br>This parameter is for Microsoft Windows only. | `service_max_native_size=131072` |
| service_min_heap_size | Specifies the minimum heap size for the adapter JVM.<br>Possible Value: A valid JVM heap size.<br>Default Value: `536870912`.<br>This parameter is for Microsoft Windows only. | `service_min_heap_size=536870912` |

*Table 2–7 (Cont.) adapter.ini Parameters*

| Parameter | Description | Example |
|---|---|---|
| service_num_vm_args | Specifies the number of service_vm_arg*number* parameters specified in JVM.<br>Possible Value: The number of service_vm_arg*number* parameters.<br>Default Value: None.<br>This parameter is for Microsoft Windows only. | service_num_vm_args=1 |
| service_path | Specifies the environment variable PATH. The PATH variable is set before starting the Java Virtual Machine (JVM). Typically, list all directories that contain necessary DLLs.<br>Possible Value: The valid PATH environment variable setting.<br>Default Value: None.<br>This parameter is for Microsoft Windows only. | service_path=%JREHOME%\bin;D:\oracle\oraic\bin |
| service_vm_arg*number* | Specifies any additional arguments to the JVM. For example, to retrieve line numbers in any stack traces, set service_vm_arg1=java.compiler=NONE. If a list of arguments exists, then use multiple parameters as shown in the example, by incrementing the last digit by 1.<br>Possible Value: A valid JVM arguments.<br>Default Value: None.<br>This parameter is for Microsoft Windows only. | service_vm_arg1=java.compiler=NONE<br>service_vm_arg2=oai.adapter=.aq |

Table 2–8 shows the reserved characters used to specify the value of the nls_date_format parameter. Use these characters to define date formats.

*Table 2–8 Reserved Characters for the nls_date_format Parameter*

| Letter | Description | Example |
|---|---|---|
| G | Era designator | AD |
| y | Year | 1996 or 96 |
| M | Month in year | July or Jul or 07 |
| w | Week in year | 27 |
| W | Week in month | 2 |
| D | Day in year | 189 |
| d | Day in month | 10 |
| F | Day of week in month | Number 2 |
| E | Day in week | Tuesday or Tue |
| a | a.m./p.m. marker | P.M. |
| H | Hour in day (0-23) | 0 |
| k | Hour in day (1-24) | 24 |
| K | Hour in a.m./p.m. (0-11) | 0 |
| h | Hour in a.m./p.m. (1-12) | 12 |
| m | Minute in hour | 30 |

*Table 2–8 (Cont.) Reserved Characters for the nls_date_format Parameter*

| Letter | Description | Example |
|---|---|---|
| s | Second in minute | `55` |
| S | Millisecond | `978` |

### FTP Adapter-specific Parameters

Table 2–9 lists the parameters specific to the FTP adapter. With the exception of the `bridge_class` parameter, all parameters can be edited after installation.

*Table 2–9 FTP Adapter-specific Parameters*

| Parameter | Description | Example |
|---|---|---|
| bridge_class | Specifies the entry class for the FTP adapter. A value must be specified and cannot be modified later.<br>Possible Value: `oracle.oai.agent.adapter.technology.TechBridge`.<br>Default Value: None. | `bridge_class=oracle.oai.agent.adapter.technology.TechBridge` |
| file.receiver.max_msgs_retrieved | Defines the maximum number of messages to be retrieved in each session.<br>Default Value: `30`. | `file.receiver.max_msgs_retrieved = 10` |
| file.receiver.polling_interval | Defines the time interval to poll the message source in milliseconds.<br>Default Value: 60000. | `file.receiver.polling_interval = 10000` |
| file.receiver.proxy_host | Specifies the name of the computer that servers as the proxy server for the inbound FTP server.<br>Possible Value: A correct host name.<br>Default Value: None. | `file.receiver.proxy_host=www-proxy.foo.com` |
| file.receiver.user | Specifies the FTP user name for the inbound FTP server.<br>Possible Value: A valid FTP user name.<br>Default Value: None. | `file.receiver.user=joe` |
| file.sender.customizer_class | Specifies the class name for customization used by the file sender.<br>Default Value: `oracle.oai.agentt.adapter.technology.FileDefaultSenderCustomizer` | `file.sender.customizer_class=MyFileSenderCustomizer` |
| file.sender.file_name_rule | Specifies the rule for generating file names used by the file sender.<br>Default Value: `%APP%%PART%_%TIME%` | `file.sender.file_name_rule=%APP%_%EVENT%_%TIME%.xml` |
| file.sender.proxy_host | Specifies the name of the computer that serves as the proxy server for the outbound FTP server.<br>Possible Value: A correct host name.<br>Default Value: None. | `file.sender.proxy_host=www-proxy.foo.com` |

*Table 2–9 (Cont.) FTP Adapter-specific Parameters*

| Parameter | Description | Example |
|---|---|---|
| file.sender.proxy_port | Specifies the port number of the proxy server for the outbound FTP server.<br>Possible Value: A valid port number.<br>Default Value: None. | `file.sender.proxy_port=80` |
| file.sender.staging_dir | Specifies the staging directory name for the file sender.<br>Default Value: None. | `file.sender.staging_dir=/private/ipdev1/ftpstaging1` |
| file.sender.type | Indicates the file types.<br>Possible Value: ASCII or binary.<br>Default Value: binary. | `file.sender.type=ASCII` |
| file.sender.user | Specifies the FTP user name for the outbound FTP server.<br>Possible Value: A valid FTP user name.<br>Default Value: None. | `file.sender.user=joe` |
| ota.d3ls | Specifies the list of data definition description language (D3L) XML files used by the bridge. Each business event handled by the bridge must have its own D3L XML file. Whenever a new D3L XML file is imported in iStudio for use by an application using the FTP adapter, the parameter must be updated and the FTP adapter restarted.<br>Default Value: None. | `ota.d3ls=person.xml, person1.xml` |
| ota.receive.endpoint | Defines the FTP receiving endpoint URL. The URL is written as follows: `ftp://host name/directory path` or `file://localhost/directory path`.<br>Possible Value: `ftp://host name/directory path`.<br>Default Value: None.<br>On Windows platforms, if you have a file system endpoint, then you can use the drive letter and directory name as part of the endpoint URL. | `ota.receive.endpoint= ftp://ip-sun/private/ipdev1/test/inbound`<br>On Windows:<br>`ota.receive.endpoint=file://localhost/c:\test` |
| ota.send.endpoint | Defines the FTP sending endpoint URL. The URL is written as follows: `ftp://host name/directory path` or `file://localhost/directory path`.<br>Possible Value: `ftp://host name/directory path`.<br>Default Value: None.<br>**Note**: Do not set the `ota.send.endpoint` parameter to the same value as the `ota.receive.endpoint` parameter unless you perform a loop-back test for the FTP adapter. The files sent are used by the receiving end. | `ota.send.endpoint=ftp://ip-sun/private/ipdev1/test/inbound` |

*Table 2–9 (Cont.) FTP Adapter-specific Parameters*

| Parameter | Description | Example |
|---|---|---|
| ota.time_format | Specifies the format in which you want to define the timestamp.<br>If you don't want to define the smtp_sender_customizer_class parameter, but still want to format file name generation, use this parameter.<br>Possible Value: The format options are identical to `java.text.SimpleDateFormat`.<br>Default Value: None. | `ota.time_format=yyyyMMddHHmmss` |
| ota.time_format_couuntry_code | Specifies the country code.<br>If `ota.time_format` is null then other two will be ignored and timestamp will be in milliseconds. If `ota.time_format` is not null and if either this parameter or `ota.time_format_language_code` parameter is null, then default value for both the parameters is used, ignoring the values given.<br>Possible Value: A valid ISO country code as mentioned in `java.util.Locale`.<br>Default Value: None. | `ota.time_format_couuntry_code=FR` |
| ota.time_format_language_code | Specifies the language code.<br>If `ota.time_format` is null, then this parameter will be ignored and timestamp will be in milliseconds. If `ota.time_format` is not null and if either this parameter or `ota.time_format_couuntry_code` parameter is null, then default value for both the parameters is used, ignoring the values given.<br>Possible Value: A valid ISO language code as mentioned in `java.util.Locale`.<br>Default Value: None. | `ota.time_format_language_code=fr` |
| ota.type | Defines the type of payload this adapter handles.<br>Possible values: `XML` and `D3L`.<br>Default Value: None. | `ota.type=XML` |

*Table 2–9 (Cont.) FTP Adapter-specific Parameters*

| Parameter | Description | Example |
|---|---|---|
| file.receiver.exception_dir | Specifies a URL which represents either an FTP directory or a file location. An FTP URL can only be specified for the exception directory if the receiving endpoint is also an FTP URL, and the host name in the URL is the same. When a processing exception occurs, the host name, user name, and password of the receiving endpoint will be used to log on to the FTP server to store the messages that are not processed successfully. Ensure this directory exists on the FTP server (or the local file system if file URL is used) and is writable by the FTP adapter process.<br>Possible Value: A URL which represents either an FTP directory or a file location.<br>Default Value: None. | `file.receiver. execption_dir=ftp://acme.com/private/user/error`<br>or<br>`file.receiver.exception_dir=file://localhost/private/user/error` |
| file.receiver.proxy_port | Specifies the port number of the proxy server for the inbound FTP server.<br>Possible Value: A valid port number.<br>Default Value: None. | `file.receiver.proxy_port=80` |
| file.receiver.customizer_class | Specifies the class name for customization used by the file receiver.<br>Default Value: `oracle.oai.aget.adapter.technology.DefaultReceiverCustomizer` | `file.receiver.customizer_class=MyFileReceiverCustomizer` |

# 3

# Design Time and Runtime Concepts

This chapter describes the design time and runtime concepts for the FTP adapter. It contains the following topics:

- FTP Adapter Design Time Concepts
- FTP Adapter Runtime Concepts
- Customizing the FTP Adapter
- Starting the FTP Adapter
- Stopping the FTP Adapter

## 3.1 FTP Adapter Design Time Concepts

The FTP adapter can handle XML and D3L structured payloads, such as pure XML data with strings beginning with `<xml...`, and binary data described by a D3L XML file.

### 3.1.1 XML Payload Type

You can import a Document Type Definition (DTD) in iStudio, which determines how the FTP adapter parses a received XML document into an OracleAS Integration InterConnect application view event. In addition, you can use the DTD to describe how an inbound application view message is converted to an XML document. Use the message type option XML when defining a new integration point in any of the event wizards.

Ensure that the `ota.type` parameter in the `adapter.ini` file is set to `XML`, instead of `D3L`.

When the FTP adapter operates in the XML payload mode, no transformations are performed on the messages between native view and application view. Any Extensible Stylesheet Language Transformations (XSLT) should be performed either before sending an XML document to OracleAS Integration InterConnect, or after receiving one from OracleAS Integration InterConnect.

### 3.1.2 D3L Payload

The FTP adapter performs a two-way conversion and transformation of messages between application view and native format.

An application based on the FTP adapter can use the iStudio Message Type D3L and the iStudio D3L Data Type Import options when importing a datatype. In this case, messages received or sent by the FTP adapter must adhere to the fixed byte-level layout defined in a D3L XML file.

The D3L Data Type Import option can also define common view datatypes.

> **See Also:** *Oracle Application Server Integration InterConnect User's Guide*, Appendix B, for additional information on D3L

## 3.2 FTP Adapter Runtime Concepts

This section describes the key runtime components of the FTP adapter. This section contains the following topics:

- FTP Receiver
- FTP Sender

### 3.2.1 FTP Receiver

The FTP adapter can receive messages from a single receiving endpoint, which is either a remote FTP server or a local file system. The receiving endpoint is in the following form:

- For an FTP protocol: `ftp://host name/directory path`
- For a local file system: `file://localhost/directory path`

The agent converts the application view event into a common view event and passes it to OracleAS Integration InterConnect for further routing and processing. Once the message is successfully passed to OracleAS Integration InterConnect, the corresponding FTP file on the remote FTP server or local file system is removed. An exception directory on the remote FTP server or local file system can be specified for storing the unsuccessfully processed files. If no exception directory is specified, then the file will be discarded. The archive of exception files of the FTP adapter is appended with a time stamp *`original filename_timestamp`*. This is to avoid the two incoming files having the same file name. The properties for the FTP receiver are found in the `adapter.ini` file as `file.receiver.*`.

> **Note:** The value of the exception directory should be a URL which represents either an FTP directory or a file location. An FTP URL can only be specified for the exception directory if the receiving endpoint is also an FTP URL. The host name in the URL is the same. When a processing exception occurs, the host name, user name, and password of the receiving endpoint will be used to log on to the FTP server to store the messages that are not processed successfully. Ensure this directory exists on the FTP server (or the local file system if file URL is used) and is writable by the FTP adapter process.

### 3.2.2 FTP Sender

The FTP adapter is comprised of the FTP bridge and the runtime agent. When the agent has a message to send to an endpoint, the bridge is notified. The bridge then uses D3L or XML to perform the conversion of the common view object to the native format. The native format message is then sent through the FTP transport layer to a FTP endpoint. The FTP endpoint is written as follows:

```
ftp://host name/directory path
```

The file name at the destination site is automatically generated by the adapter and is in the following syntax:

```
ftp adapter name partition-time stamp
```

You can specify a rule for generating the file name when the FTP adapter sends a file to a directory or FTP server. To use this feature, add the parameter `file.sender.file_name_rule` to the `adapter.ini` file. The adapter recognizes the following tokens:

- `%APP%`: application name
- `%BO%`: business object name
- `%EVENT%`: corresponding event name
- `%MV%`: message version
- `%PART%`: partition number
- `%TIME%`: time stamp
- `%TYPE%`: message type

For example, `file.sender.file_name_rule=%APP%_%EVENT%_%TIME%.xml`. This rule instructs the adapter to generate files with the following pattern:

```
your app name_event name_current time stamp.xml
```

The FTP adapter supports sending outgoing messages from OracleAS Integration InterConnect to multiple FTP endpoints. The multiple endpoints feature enables sending messages to various FTP servers.

An endpoint can be associated with a subscribing event in iStudio by adding transport properties such as FTP endpoint, FTP user name, and endpoint password as metadata for the event. This is done using the Modify Fields function of the Subscribe Wizard - Define Application View dialog. After associating an endpoint and event, the message from the subscribing event is sent to the FTP endpoint.

When using the multiple-endpoint feature with `XML` datatype, choose the event type of `Generic`, instead of `XML`. Using the `Generic` event type allows you to enter the metadata for the endpoints using the Modify Fields feature associated with iStudio.

Table 3–1 shows how metadata is associated to an event called `SendOrder` that sends an order to a FTP server at foo.com in the `/private/user/test` directory.

***Table 3–1 SendOrder Event Metadata***

| Parameter | Description |
|---|---|
| `file.sender.user=joe` | Defines the FTP user name. |
| `ota.endpoint=sendOrderAppEP` | Specifies a unique endpoint name set in iStudio |
| `ota.send.endpoint=http://foo.com/servlet/test` | Specifies the sending endpoint for the FTP adapter |

**Note:** The `sender` properties are not inherited from the `adapter.ini` file.

**See Also:** *Oracle Application Server Integration InterConnect User's Guide*

To avoid external applications picking up partial files, files that are not completely transferred, specify a staging directory. Set the `file.sender.staging_dir` parameter in the `adapter.ini` file. This parameter should only contain the directory name, for example, `file.sender.staging_dir=/private/ipdev/test/staging`.

> **Note:** Do not use file or FTP URL, such as `ftp://...` or `file://....` If the staging directory is used for an FTP server, then the path specified is a directory on the FTP server. Ensure that the path for the staging directory is valid.

The properties for the FTP Receiver are in the `adapter.ini` file as `file.sender.*`.

## 3.3 Customizing the FTP Adapter

You can customize the adapter behavior by implementing the following interfaces:

- `oracle.oai.adapter.agent.technology.ReceiverCustomizer`
- `oracle.oai.adapter.agent.technology.FileSenderCustomizer`

### 3.3.1 ReceiverCustomizer Interface

You can use the `ReceiverCustomizer` interface to customize the `TransportMessage` object that FTP adapter receives. The `TransportMessage` object represents the native message that the transport layer receives or sends.

- If you wish to customize the `TransportMessage` object itself, then use the `customizeTransportMessage()` method. This method is called before the before the adapter processes the `TransportMessage` object.
- If you wish to modify the message itself, then implement the `customizeTransportMessage()` method. You must also implement the `createReplyMessage()` method and ensure that it returns a `null` value.

The following code described the file structure of the `ReceiverCustomizer` interface.

```
package oracle.oai.agent.adapter.technology;
import oracle.oai.agent.adapter.transport.TransportMessage;
import oracle.oai.agent.adapter.sdk.Agent;
public interface ReceiverCustomizer {

   public void customizeTransportMessage(Agent agent,
                                      int receiverType,
                                      TransportMessage transportMessage);

   public String createReplyMessage(Agent agent,
                                 int status,
                                 TransportMessage receivedTransportMessage);
}
```

The following table summarizes the `ReceiverCustomizer` Interface.

| Methods | Description |
|---|---|
| createReplyMessage(); | Creates a reply message based on the status and the message received. It contains the following parameters:<br>agent: Log a message.<br>status: Status of the message process. If the value is TransportResponse.TRANSPORT_ACK, then the message is processed successfully. If the value is TransportResponse.TRANSPORT_ERROR, then the message is processed unsuccessfully.<br>receivedTransportMessage: Transport message is received by the adapter. This parameter is used to transport headers in the transport message to create a meaningful message.<br>The return string contains the reply message. This method is used for backward compatibility for the FTP adapter. However, for the FTP adapter, you should return a null value in this method. |
| customizeTransportMessage(); | Allows you to customize the transport message received by the adapter. It uses the following parameters:<br>agent: Log a message.<br>receiverType: Information on the type of adapter.<br>transportMessage: Customize the transport message received by the adapter. |

***Example 3–1 Example of ReceiverCustomizer***

The MyReceiverCustomizer class removes the first line in the native message.

```
import oracle.oai.agent.adapter.sdk.Agent;
import oracle.oai.agent.adapter.transport.TransportMessage;
import oracle.oai.agent.adapter.transport.TransportException;
import oracle.oai.agent.adapter.technology.ReceiverCustomizer;

public class MyReceiverCustomizer implements ReceiverCustomizer {
```

This example describes how to remove an extra line in the incoming files.

```
   public void customizeTransportMessage(Agent agent, int receiverType,
                                         TransportMessage transportMessage)
   {
      String payload = transportMessage.getBodyAsString();

      agent.logTraceMessage("payload received = " + payload, null, null, null);
//the following syntax removes the first line from the payload.
      String newPayload = removeFirstLine(payload);

      try {
         transportMessage.setBody(newPayload);
      }
      catch(TransportException te) {
      . . . .
      }
   }

   public String createReplyMessage(Agent agent, int status,
                                    TransportMessage receivedTransportMessage)
   {
   return null;
   }
}
```

### List of Methods for the TransportMessage Class

The following table provides a list of methods you may choose for the `TransportMessage` class.

| Method | Description |
|---|---|
| public byte[] getBodyAsBytes(); | Get the body of the message as byte array. Return the message in byte[]. |
| public InputStream getBodyAsInputStream(); | Get the body of the message and return an InputStream object representing the body of the message. |
| public Properties getTransportHeaders(); | Get all transport-specific headers and return a Properties object that contains all the transport headers. |
| public String getBodyAsString(); | Get the body of the message as String object. Return the message in String object. |
| public String toString(); | Dump messages and headers. |
| public void setBody(InputStream in) throws TransportException; | Set the body of the message. The body type will be set to BYTES. Parameter includes:<br>InputStream: Contains the message.<br>It throws a TransportException. |
| public void setBody(String body) throws TransportException; | Set the body of the message. The body type will be set to STRING. Parameter includes:<br>body: body of the message<br>It throws a TransportException. |

## 3.3.2 FileSenderCustomizer Interface

You can use the `FileSenderCustomizer` interface to customize the file name and payload of the `TransportMessage` object that is sent to the transport layer.

The `FileSenderCustomizer` interface extends the `SenderCustomizer` interface. You must implement the `FileSenderCustomizer` interface by implementing the following two methods:

- `FileSenderCustomizer.customizeTransportMessage()`
- `FileSenderCustomizer.generateFileName()`

If you do not want to implement the `generateFileName()` method, then you can create a class that extends the `oracle.oai.agent.adapter.technology.FileDefaultSenderCustomizer` class, which is provided in the `oai.jar` file. In this case, you only need to implement the `customizeTransportMessage()` method.

### 3.3.2.1 SenderCustomizer Interface

The following code describes the file structure of the `SenderCustomizer` interface.

```
package oracle.oai.agent.adapter.technology;

import oracle.oai.agent.adapter.sdk.MessageObject;
import oracle.oai.agent.adapter.sdk.AttributeObject;
import java.util.Properties;
import oracle.oai.agent.adapter.sdk.Agent;
import oracle.oai.agent.adapter.transport.TransportMessage;

public interface SenderCustomizer {
   public void customizeTransportMessage(Agent agent,
                                        TransportMessage transportMessage,
                                        MessageObject mobj,
                                        AttributeObject aobj);
}
```

### customizeTransportMessage method

This method specifies how to customize the transport message for transporting sender. The adapter creates a `TransportMessage` object for the transport layer to send, based on the `MessageObject` object sent by OracleAS Integration InterConnect.

This method contains the following parameters:

`agent`: Log messages.

`transportMessage`: The `TransportMessage` object that the adapter has created for sending.

`mobj`: The `MessageObject` from OracleAS Integration InterConnect.

`aobj`: The `AttributeObject` from OracleAS Integration InterConnect.

This method does not return anything. You can change the payload with the `transportMessage` parameter.

## 3.3.2.2 FileSenderCustomizer Interface

The following code describes the file structure of the `FileSenderCustomizer` interface.

```
package oracle.oai.agent.adapter.technology;
import java.util.Date;
import oracle.oai.agent.adapter.sdk.MessageObject;
import oracle.oai.agent.adapter.sdk.AttributeObject;
import oracle.oai.agent.adapter.sdk.Agent;

public interface FileSenderCustomizer extends SenderCustomizer {
   public String generateFileName (Agent agent,
                                   String rule,
                                   String app,
                                   String partitition,
                                   Date time,
                                   MessageObject mobj,
                                   AttributeObject aobj);
}
```

### generateFileName method

This method generates a file name for sending a file. It contains the following parameters:

`agent`: Use the `Agent` object to log message.

`rule`: Rule for generating subject. This parameter is read from `file.sender.file_name_rule` in `adapter.ini`.

`app`: The application name.

`partition`: Partition.

`time`: The time the OracleAS Integration InterConnect object is received.

`mobj`: A `MessageObject` passed from OracleAS Integration InterConnect.

aobj: An AttributeObject passed from OracleAS Integration InterConnect.

This method returns a string representing the file name.

## 3.4 Starting the FTP Adapter

The process for starting the adapter varies based on the operating system.

- To start the FTP adapter on UNIX:
  1. Change to the directory containing the start script.

     ```
     cd ORACLE_HOME/integration/interconnect/adapters/Application
     ```

  2. Type **start** and press **Enter**.
- To start the FTP adapter from Services on Windows:
  1. Access the Services window from the Start menu. The Services window is displayed.

| On... | Choose... |
|---|---|
| Windows 2000 | Start, Settings, Control Panel, Administrative Tools, Services |

  2. Select the **OracleHomeOracleASInterConnectAdapter-Application** service.
  3. Start the service based on the operating system.

| On... | Choose... |
|---|---|
| Windows 2000 | Right-click the service and choose Start from the context menu. |

> **Note:** You can also start and stop the FTP adapter using the IC Manager. Refer to *Oracle Application Server Integration InterConnect User's Guide* for more details.

### 3.4.1 Log File of FTP Adapter

You can verify the start up status of the FTP adapter by viewing the oailog.txt files. The files are located in the timestamped subdirectory of the log directory of the FTP adapter. Subdirectory names have the following form:

```
timestamp_in_milliseconds
```

The following is an example of the information about an FTP adapter that started successfully:

```
The Adapter service is starting..
Registering your application (FTPAPP)..
Initializing the Bridge oracle.oai.agent.adapter.technology.TechBridge..
Starting the Bridge oracle.oai.agent.adapter.technology.TechBridge..
Service started successfully.
```

## 3.5 Stopping the FTP Adapter

The process for stopping the adapter varies based on the operating system.

- To stop the FTP adapter on UNIX:

  1. Change to the directory containing the stop script.

     ```
     cd ORACLE_HOME/integration/interconnect/adapters/Application
     ```

  2. Type **`stop`** and press **Enter**.

- To stop the FTP adapter from Services on Windows:

  1. Access the Services window from the Start menu. The Services window is displayed.

| On... | Choose... |
|---|---|
| Windows 2000 | Start, Settings, Control Panel, Administrative Tools, Services |

  2. Select the **OracleHomeOracleASInterConnectAdapter-Application** service.
  3. Stop the service based on the operating system.

| On... | Choose... |
|---|---|
| Windows 2000 | Right-click the service and choose Stop from the context menu. |

You can verify the stop status of the FTP adapter by viewing the `oailog.txt` files. These files are located in the timestamped subdirectory of the `log` directory of the FTP adapter.

# A

# Frequently Asked Questions

This appendix provides answers to frequently asked questions about the FTP adapter.

- How do I know the FTP Adapter has started properly?
- The FTP Adapter did not start properly. What went wrong?
- Is it possible to edit the FTP adapter configuration settings created during installation?
- When I change an element in iStudio, such as mappings, it seems like the FTP Adapter is using old information. What is happening?
- How do I secure my passwords?

## How do I know the FTP Adapter has started properly?

View the `oai.txt` file located in the timestamped subdirectory of the FTP adapter log directory:

| Platform | Directory |
|---|---|
| UNIX | *ORACLE_HOME*/integration/interconnect/adapters/*Application*/log/*timestamp_in_milliseconds* |
| Windows | *ORACLE_HOME*\integration\interconnect\adapters\*Application*\log\*timestamp_in_milliseconds* |

If there are no exceptions, then the FTP adapter has started properly.

## The FTP Adapter did not start properly. What went wrong?

View the exceptions in the FTP adapter log file (`oailog.txt`). The exceptions should provide information about what went wrong. It is possible that the FTP adapter is unable to connect to the repository. Ensure the repository is started properly. The FTP adapter will connect to the repository once it is started properly. You do not need to restart the Adapter.

> **See Also:** *Oracle Application Server Integration InterConnect User's Guide* for instructions on starting the repository on UNIX and Windows

## Is it possible to edit the FTP adapter configuration settings created during installation?

Yes, edit the parameters in the `adapter.ini` file in the following directory:

| Platform | Directory |
|---|---|
| UNIX | *ORACLE_HOME*/integration/interconnect/adapters/*Application*/ |
| Windows | *ORACLE_HOME*\integration\interconnect\adapters\*Application*\ |

**See Also:** Chapter 2, "Installation and Configuration"

### When I change an element in iStudio, such as mappings, it seems like the FTP Adapter is using old information. What is happening?

The FTP adapter caches information from iStudio. The information is stored in the repository locally. If you change something in iStudio and want to view the change in the runtime, then you need to stop the FTP adapter, delete the FTP adapter cache files, and restart the FTP adapter.

The FTP adapter has a persistence directory which is located in the FTP adapter directory. Deleting this directory when the FTP adapter has been stopped should make it obtain the new metadata from the repository when started.

### How do I secure my passwords?

OracleAS Integration InterConnect uses Oracle Wallet Manager to maintain system passwords. When you install OracleAS Integration InterConnect, Oracle Wallet Manager is also installed and a password store is created. All passwords used by OracleAS Integration InterConnect components are stored in the password store. The password is stored in the Oracle Wallet in the following format:

```
ApplicationName/password
```

For example,

```
AQAPP/aq_bridge_schema_password
```

The `ApplicationName` is the name of the application, which is extracted from the `adapter.ini` file of the corresponding adapter. In the `adapter.ini` file, the `application` parameter specifies the `ApplicationName` to which this adapter connects. The password for the application is also retrieved from the `adapter.ini` file.

You can create, update, and delete passwords using the `oraclewallet` command. When you run the command, it prompts you for the admin password.

You can use the following commands to manage your passwords:

- List all passwords in the store

  ```
  oraclewallet -listsecrets
  ```

- Create a password

  ```
  oraclewallet -createsecret passwordname
  ```

  For example, to create a password for the hub schema:

  ```
  oraclewallet -createsecret hub_password
  ```

- View a password

  ```
  oraclewallet -viewsecret passwordname
  ```

  For example, to view the password for the hub schema:

  ```
  oraclewallet -viewsecret hub_password
  ```

- Update a password

  ```
  oraclewallet -updatesecret passwordname
  ```

  For example, to update the password for the hub schema:

  ```
  oraclewallet -updatesecret hub_password
  ```

- Delete a password

  ```
  oraclewallet -deletesecret passwordname
  ```

  For example, to delete the password for the hub schema:

  ```
  oraclewallet -deletesecret hub_password
  ```

# B

# Example of the adapter.ini File

This appendix shows a sample `adapter.ini` file for the FTP adapter.

> **See Also:** Configuring the FTP Adapter on page 2-6 for additional information on `adapter.ini` configuration parameters

The following code is an example of the FTP `adapter.ini` file.

```
#include <../../hub/hub.ini>

// *************
// ** Adapter **
// *************

// Application (as created in iStudio) corresponding to this Adapter.
application=myFtpApp

// Partition (as created in iStudio) corresponding to this Adapter.
partition=

// If you have multiple adapter instances for a given application with the
// given partition, each Adapter should have an instance number.

//instance_number=2

// Bridge class
bridge_class=oracle.oai.agent.adapter.technology.TechBridge

ota.type=D3L


// define the ftp sending endpoint
// For ftp, ota.send.endpoint=ftp://host name/path name
// For file, ota.send.endpoint=file://host name/path name
//
ota.send.endpoint= ftp://foo.s.com/private/ipdev1/test/d3l/inbound

// define the ftp receiving endpoint
// For ftp, ota.send.endpoint=ftp://host name/path name
// For file, ota.send.endpoint=file://host name/path name
//
ota.receive.endpoint=ftp://foo.s.com/private/ipdev1/test/d3l/inbound


//-------------------------------------
// ftp Sender initialization variables
```

```
//-------------------------------------

// ftp user (mandatory if ftp is used)
// file.sender.user=ipdev1
file.sender.user=ipdev1


// ftp user password (mandatory if ftp is used)
//file.sender.password=ipdev1
file.sender.password=ipwelcome


// file type (ASCII or BINARY)
//file.sender.type=BINARY
file.sender.type=ASCII

// proxy host
//file.sender.proxy_host=

// proxy port
//file.sender.proxy_port=
//staging directory
//file.sender.staging_directory =/tmp

//sender customizer class
//file.sender.customizer_class = MySenderCustomizer

//-------------------------------------
// ftp receiver initialization variables
//-------------------------------------

// ftp user (mandatory if ftp is used)
//file.receiver.user=ipdev1
file.receiver.user=ipdev1

// ftp user password (mandatory if ftp is used)
//file.receiver.password=ipdev1
file.receiver.password=ipwelcome

// file type (ASCII or BINARY)
//file.receiver.type=BINARY
file.receiver.type=BINARY

// proxy host
//file.receiver.proxy_host=

// proxy port
//file.receiver.proxy_port=
//receiver customizer class
//file.receiver.customizer_class = MyReceiverCustomizer

// define where to put the file that cannot be processed properly.
//file.receiver.exception_dir=


// define how often to poll
// the message source (in milli seconds)
file.receiver.polling_interval=60000

// define maximum number of messages
// retrieved in each polling session
file.receiver.max_msgs_retrieved=30

// D3L initialization variables
ota.d3ls=person2.xml:person1.xml


// *************
// ** Agent  ***
// *************

// Log level (0 = errors only, 1 = status and errors, 2 = trace, status and
// errors).
agent_log_level=2

// Hub message selection information
agent_subscriber_name=myFTPApp
agent_message_selector=recipient_list like '%,myFTPApp,%'
// Only provide values for the next two parameters if you have multiple Adapter
// instances for the given application with the given partition.

//agent_reply_subscriber_name=
//agent_reply_message_selector=

// Set this to false if you want to turn off all tracking of messages (if true,
// messages which have tracking fields set in iStudio will be tracked)

agent_tracking_enabled=true

// Set this to false if you want to turn off all throughput measurements
agent_throughput_measurement_enabled=true

// By default, Adapters use an OAI specific DTD for all messages sent to the Hub
//as other OAI Adapters will be picking up the messages from the Hub and know
// how to interpret them. This should be set to true if for every message, you
//would like to use the DTD imported for that message's Common View instead
//of the OAI DTD. This should only be set to true if an OAI Adapter
//is *NOT* receiving the messages from the Hub.

agent_use_custom_hub_dtd=false

// Sets the metadata caching algorithm. The possible choices are startup (cache
everything at startup: this may take a while if there is a lot of metadata in
// your Repository), demand (cache metadata as it is used) or none (no caching: //
this will slow down performance.)
agent_metadata_caching=demand

// Sets the DVM table caching algorithm. The possible choices are startup (cache
// all DVM tables at startup: this may take a while if there are a lot of tables
// in your Repository), demand (cache tables as they are used) or none (no caching
//: this will slow down performance.)
agent_dvm_table_caching=demand

// Sets the lookup table caching algorithm. The possible choices are startup
// (cache all lookup tables at startup: this may take a while if there are a lot
// of tables in your Repository), demand (cache tables as they are used) or none
// (no caching: this will slow down performance.)
agent_lookup_table_caching=demand

// If metadata caching, DVM table caching, or lookup table caching are turned on
```

```
//(startup or demand) then the Adapter caches metadata or DVM tables it retrieves
//from the Repository in a file cache. When you restart the Adapter,it will not
// have to get that metadata or DVM table from the Repository again because it is
// in the cache files.However, if you change some metadata or DVM table using
// iStudio and you want the Adapter to use those changes the next time it is
// started you can either delete the cache files or set this parameter to true
// before restarting.
agent_delete_file_cache_at_startup=false

// Max number of application datatype information to cache
agent_max_ao_cache_size=200

// Max number of common datatype information to cache
agent_max_co_cache_size=100

// Max number of message metadata to cache
agent_max_message_metadata_cache_size=200

// Max number of DVM tables to cache
agent_max_dvm_table_cache_size=200

// Max number of lookup tables to cache
agent_max_lookup_table_cache_size=200

// Internal Agent queue sizes
agent_max_queue_size=1000
agent_Persistence_queue_size=1000

// Persistence
agent_persistence_cleanup_interval=60000
agent_persistence_retry_interval=60000


 //////////////////
 // End Comments //
//////////////////
```

# Index