Oracle9*i*AS InterConnect Adapter for FTP

Installation and User's Guide

Release 2 (9.0.2)

February 2002
Part No. A95440-01

ORACLE®

Oracle9*i*AS InterConnect Adapter for FTP Installation and User's Guide, Release 2 (9.0.2)

Part No. A95440-01

Contents

1   Introduction

2   Installation and Configuration

3   Design Time and Runtime Concepts

4   Frequently Asked Questions

A   Sample Adapter.ini File

Index

Send Us Your Comments

Oracle9*i*AS InterConnect Adapter for FTP Installation and User's Guide, Release 2 (9.0.2)

Part No. A95440-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. 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?

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

- Electronic mail - iasdocs_us@oracle.com
- Fax - (650) 506-7407 Attn: Oracle9*i* Application Server Documentation Manager
- Postal service:
  > Oracle Corporation
  > Oracle9*i* Application Server Documentation Manager
  > 500 Oracle Parkway, M/S 2op3
  > Redwood Shores, CA 94065 USA

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

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

Preface

This preface contains these topics:

- Audience
- Documentation Accessibility
- Organization
- Related Documentation
- Conventions

Audience

*Oracle9i*AS InterConnect Adapter for FTP Installation and User's Guide is the primary source of introduction, installation, and usage information for the File Transfer Protocol (FTP) adapter. This preface discusses the following topics:

- Audience

- Documentation Accessibility

- Organization

- Related Documentation

- Conventions

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 Corporation 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.

Organization

This document contains:

Chapter 1, "Introduction"
This chapter describes the 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 for the FTP adapter.

Chapter 4, "Frequently Asked Questions"

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

Related Documentation

For more information, see these Oracle resources:

- Oracle9*i*AS InterConnect User's Guide in the Oracle9i Application Server Documentation Library

- Oracle9*i*AS InterConnect Adapter Configuration Editor User's Guide

In North America, printed documentation is available for sale in the Oracle Store at

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

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

`http://www.oraclebookshop.com/`

Other customers can contact their Oracle representative to purchase printed documentation.

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://otn.oracle.com/admin/account/membership.html`

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

`http://otn.oracle.com/docs/index.htm`

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 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.	*Oracle9i Database Concepts*
		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, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a `NUMBER` column.
		You can back up the database by using the `BACKUP` command.
		Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view.
		Use the `DBMS_STATS.GENERATE_STATS` procedure.

Convention	Meaning	Example
`lowercase monospace (fixed-width font)`	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to open SQL*Plus. The password is specified in the `orapwd` file. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table. Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true`. Connect as `oe` user. The `JRepUtil` class implements these methods.
`lowercase monospace (fixed-width font) italic`	Lowercase monospace italic font represents placeholders or variables.	You can specify the `parallel_clause`. Run `Uold_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
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (digits [ , precision ])`
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE	DISABLE}`
|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	`{ENABLE	DISABLE}`  `[COMPRESS	NOCOMPRESS]`

Convention	Meaning	Example
`...`	Horizontal ellipsis points indicate either:	
	■ That we have omitted parts of the code that are not directly related to the example	`CREATE TABLE ... AS` *`subquery;`*
	■ That you can repeat a portion of the code	`SELECT` *`col1, col2,`* `... ,` *`coln`* `FROM employees;`
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	`acctbal NUMBER(11,2);` `acct    CONSTANT NUMBER(4) := 3;`
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/`*`system_password`* `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. However, because these terms are not case sensitive, you can enter them in lowercase.	`SELECT last_name, employee_id FROM employees;` `SELECT * FROM USER_TABLES;` `DROP TABLE hr.employees;`
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	`SELECT last_name, employee_id FROM employees;` `sqlplus hr/hr` `CREATE USER mjones IDENTIFIED BY ty3MU9;`

1

Introduction

This chapter discusses the FTP adapter and the hardware and software requirements. This chapter discusses the following topics:

- FTP Adapter Overview

FTP Adapter Overview

The FTP adapter enables an Oracle FTP Application to be integrated with other applications using Oracle9*i*AS InterConnect. This adapter is useful in all enterprise application integration scenarios involving the FTP transport protocol or local file system. 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 via proxy host. The payload for this adapter can be XML data or D3L data.

The following two diagrams show the data flow of incoming messages, from remote FTP server/local file system to Oracle9*i*AS InterConnect, and the outgoing messages from Oracle9*i*AS InterConnect to the remote FTP Server.

> **See Also:** *Oracle9iAS InterConnect User Guide*

Figure 1–1 Incoming Messages Diagram
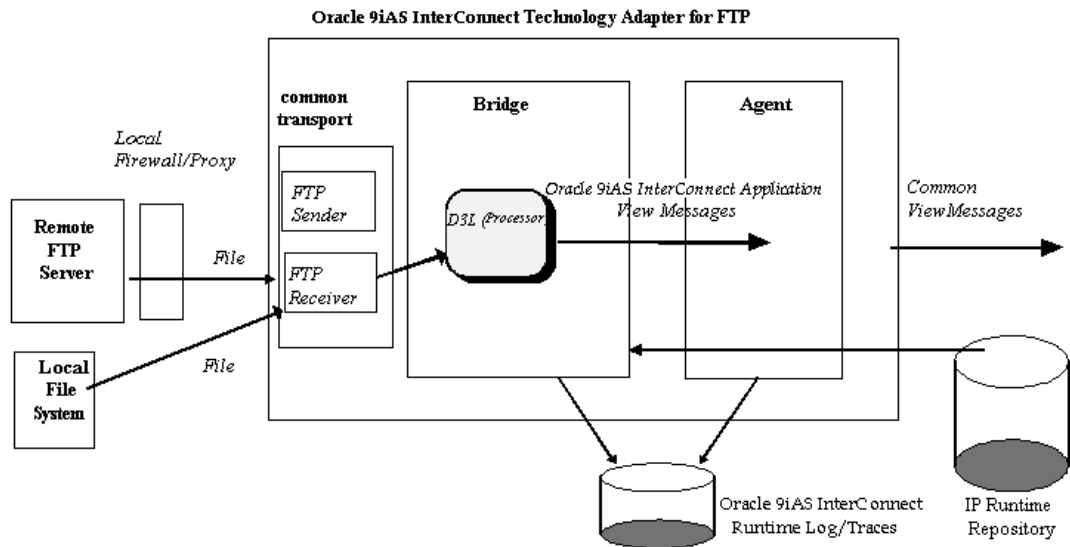
Figure 1–2   Outgoing Messages Diagram
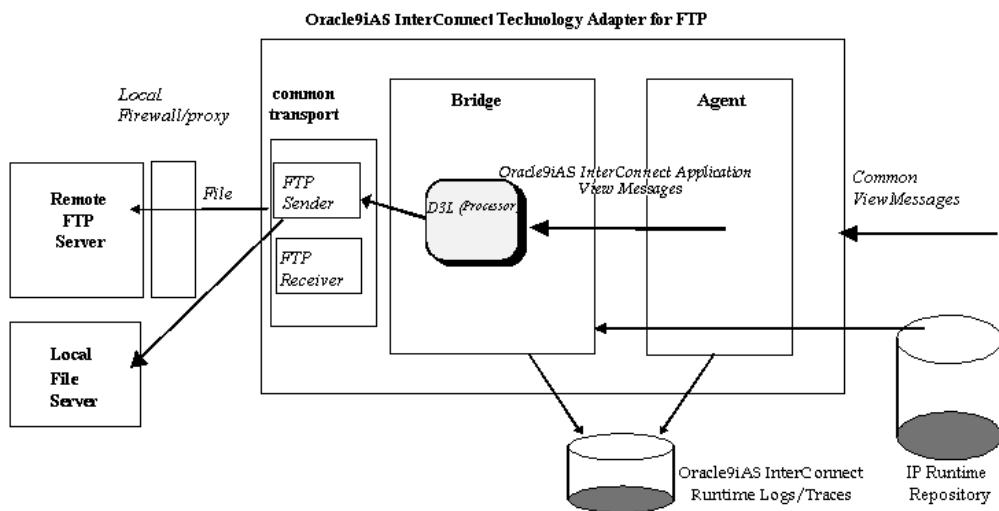
Hardware Requirements

Table 1–1 lists the hardware requirements for the computer on which the Oracle FTP adapter is installed.

Table 1–1   Hardware Requirements

Hardware	Windows NT/2000	UNIX
Memory	128 MB	128 MB
Service Pack	4.0 Service Pack 6 or later	N/A
Disk Space	500 MB	500 MB

Software Requirements

The following are software requirements for the FTP adapter:

- Operating System Requirements

- JRE Requirements

Operating System Requirements

Table 1–2 lists operating system requirements for the computer on which the FTP adapter is installed.

Table 1–2   Operating System Requirements

Operating System	Version
Windows NT/2000	version that supports at least 400 MHz
IBM AIX-Based Systems	version 4.3.3
Compaq Tru64 UNIX	version 5.0a or 5.1
HP 9000 Series HP-UX	version 11.0
SUSE LINUX SLES7	version 7.2, Kernel 2.4.7, GlibC 2.2.2-55
Sun SPARC Solaris	version 2.6

JRE Requirements

Oracle9*i*AS InterConnect uses JRE 1.3 which is installed with its components.

Known Limitations

The FTP adapter has the following limitations:

- Does not support secure FTP server.

- 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 the FTP adapter installation and configuration.

This chapter discusses these topics:

-
-

Installing the FTP Adapter

This section contains these topics:

- Preinstallation Tasks
- Installation Tasks
- Postinstallation Tasks

Preinstallation Tasks

The FTP adapter must be installed in one of the following Oracle homes:

- An existing Oracle9*i* Application Server Oracle home
- An existing Oracle9*i* Application Server Infrastructure Database Oracle home
- An existing Oracle9*i*AS InterConnect Oracle home
- A new Oracle home (the installer creates this for you)

Consult the *Oracle9i Application Server Installation Guide* before proceeding with FTP adapter installation. This guide includes information on:

- CD-ROM mounting
- Oracle Universal Installer startup
- Oracle9*i*AS InterConnect software, hardware, and system requirements
- Oracle9*i*AS InterConnect installation

> **Note:** Oracle9*i*AS InterConnect Hub is installable through the Oracle9*i*AS InterConnect Hub installation type. You must install the Oracle9*i*AS InterConnect Hub before proceeding with FTP adapter installation.

Installation Tasks

To install the FTP adapter:

1. Click Next on the Welcome page.

    The File Locations page displays.

2. Enter the following information in the Destination fields:

    ■   Name—The Oracle home name.

    ■   Path—The full path to the Oracle home in which to install the FTP adapter.

    > **Note:**   Do not change the path specified in the Source field. This is the location on the CD-ROM from which to install the FTP adapter.

3. Click Next.

    The Installation Types page displays.

4. Select Oracle9*i*AS InterConnect Adapters and click Next.

    The Available Product Components page displays.

5. Select Oracle9*i*AS InterConnect FTP Adapter and click Next.

6. If the FTP adapter is not being installed on the same computer as Oracle9*i*AS InterConnect and another adapter is not installed in the current Oracle home, the Oracle9*i*AS InterConnect Hub Database page displays. Enter the following information about the Oracle9*i*AS InterConnect Hub to use:

    ■   Host Name—The hostname of the computer on which Oracle9*i*AS InterConnect is installed

    ■   Port Number—The port number of the computer

    ■   Database SID—The system identifier (SID) of the Oracle9*i*AS InterConnect Oracle9*i*AS Metadata Repository

    ■   Password—The password for the Oracle9*i*AS Metadata Repository schema

    The Oracle9*i*AS Metadata Repository stores metadata used by Oracle9*i*AS InterConnect to coordinate communication between components.

7. Click Next.

    The Oracle9*i*AS InterConnect FTP Adapter Configuration page displays.

8. Enter the name of the application associated with the FTP adapter. White spaces or blanks are not allowed. This is the same application name created or to be created in iStudio. The default value is myFTPApp.

9. Click Next.

The Oracle9*i*AS InterConnect FTP Adapter usage page displays.

10. Select one of the following options and go to the step specified to enable the sending and/or receiving of messages from an external data source, such as an FTP server. You can change your selections later by editing parameter settings in the `adapter.ini` file.

If You Select...	Then Click Next and Go to Step...
Configure for both sending and receiving messages	11
Configure for sending messages ONLY	11
Configure for receiving messages ONLY	13

11. Enter the following information in the Oracle9*i*AS InterConnect FTP Adapter Configuration - Configure sending endpoint information page:

- Username—The username for the FTP server.

- Password—The 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, the user name, password, and file type are not required.

  ---

12. Click Next.

The installation page that displays next is based on the selection you made in Step 10:

If You Selected...	Then Go to Step...
Configure for both sending and receiving messages	13
Configure for sending messages ONLY	15

13. Enter the following information in the Oracle9*i*AS InterConnect FTP Adapter Configuration - Configure receiving endpoint information page:

- Username—The username account of the FTP server from which the Oracle9*i*AS InterConnect Hub receives messages

- Password—The password for the username 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, the user name, password, and file type are not required.

    Caution: For testing purposes, do not specify a personal FTP account or personal file directories as the receiving endpoint. 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 Corporation recommends that you create a dedicated FTP account or user a account (if the local file system is used for the receiving endpoint) for testing and deploying this adapter.

    ---

14. Click Next.

15. Complete any other fields for other components selected for installation, such as other adapters.

When finished, the Summary page displays.

16. Click Install to install the FTP adapter. The adapter is installed in the following directory:

Platform	Directory
Windows	`%ORACLE_HOME%\oai\9.0.2\adapters\`*Application*
UNIX	`$ORACLE_HOME/oai/9.0.2/adapters/`*Application*

Application is the value you specified in Step 8 on page 2-4.

Postinstallation Tasks

FTP adapter installation creates an `adapter.ini` file that consists of configuration parameters read by the FTP adapter at startup. These configuration parameter settings are appropriate for most FTP application environments. You can customize some `adapter.ini` file parameter settings for the FTP application after installation. See the following sections:

- Customizing the Payload Data Type
- Customizing the Sending Endpoints
- Customizing the Receiving Endpoints

> **See Also:**
>
> - Table 2–1 on page 2-9 for the location of the `adapter.ini` file
> - Table 2–6 on page 2-16 for `adapter.ini` file parameter setting information specific to the FTP adapter

Customizing the Payload Data Type

Payload data is the data sent between applications. If you want to change the payload data type 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`. For example:

```
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. For example:

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

    See Also: The following parameter descriptions for additional information:

    -   `ota.type` on page 2-16

    -   `ota.d3ls` on page 2-16

Customizing the Sending Endpoints

If you want to customize the behavior of the sending endpoints (destinations) for messages, edit the following parameters in the `adapter.ini` file. These parameters are not automatically set to default values during installation.

1. Change the sender endpoint by editing 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, 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, enter the appropriate values for the `file.sender.proxy_host` and `file.sender.proxy_port` parameters.

Customizing the Receiving Endpoints

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

1. Change the receiver endpoint by editing the `ota.receive.endpoint` 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`

    > **Note:**   If the endpoint is a local file system leave the
    > `file.receiver.file_type`, `file.receiver.password`,
    > `file.receiver.proxy_host`, and `file.receiver.proxy_`
    > `port` parameters blank.

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, enter the appropriate 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 and cannot be processed successfully. For example:

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

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

    ```
    file.receiver.polling_interval=20000
    ```

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
    ```

FTP Adapter Configuration

Table 2–2, Table 2–3, and Table 2–4 describe FTP executable files, configuration files, and directories. These files and directories are accessible from the directory shown in Table 2–1.

Table 2–1   FTP Adapter Directory

On...	Go to...
UNIX	`$ORACLE_HOME/oai/9.0.2/adapters/`*Application*
Windows	`%ORACLE_HOME%\oai\9.0.2\adapters\`*Application*

Table 2–2   FTP Executable Files

File	Description
`start.bat` (Windows) `start` (UNIX)	Takes no parameters; starts the adapter
`stop.bat` (Windows) `stop` (UNIX)	Takes no parameters; stops the adapter
`ignoreerrors.bat` (Windows) `ignoreerrors` (UNIX)	If an argument is specified, then the given error code is ignored: `ignoreerrors errorCodeToBeIgnored` If no argument is specified, then all error codes specified in the `ErrorCodes.ini` file are ignored: `ignoreerrors`

See Also: "FTP Error Code" on page 3-7 for a list of error codes

Table 2–3   FTP Configuration Files

File	Description
`ErrorCodes.ini` (Windows and UNIX)	Should contain one error code per line
`adapter.ini` (Windows and UNIX)	Consists of all the initialization parameters that the adapter reads at startup

See Also: Appendix A, "Sample Adapter.ini File"

Table 2–4   FTP Directories

Directory	Description
persistence	The messages are persisted (made available) in this directory. Do not edit this directory or its contents.
logs	The logging of adapter activity is done in subdirectories of the logs directory. Subdirectory names take the following form:
	timestamp_in_milliseconds
	Each time the adapter is run, a new subdirectory is created in which logging is done in an oailog.txt file.

Using the Application Parameter

Adapters do not have integration logic. The FTP adapter has a generic transformation engine that processes metadata from the repository as runtime instructions to do transformations. The application defines for an adapter what its capabilities are. For example, it can define what messages it can publish, what messages it can subscribe to, and what are the transformations to perform. The application parameter allows the adapter to become smart in the context of the application to which it is connected. It allows the adapter to retrieve from the repository only that metadata that is relevant to the application. The application parameter must match the corresponding application that will be defined in *i*Studio under the Applications folder.

If you are using pre-packaged metadata, after importing the pre-packaged metadata into the repository, start up *i*Studio to find the corresponding application (under the Applications folder in *i*Studio) to use as the application for the adapter you are installing (unless the package you are using provides directions for what the application should be).

adapter.ini Initialization Parameter File

This section contains these topics:

- Agent Connection Parameters
- FTP Adapter-Specific Parameters

Agent Connection Parameters

The agent component of the FTP adapter reads the adapter.ini file at runtime to access FTP adapter parameter configuration information. Table 2–5 lists the

parameter name, a description for each parameter, the possible and default values, and an example.

Table 2–5   Agent Connection Parameters

Parameter	Description	Example
application	Specifies the name of the application to which this adapter connects. This must match with the name specified in iStudio during creation of metadata. Use any alphanumeric string. There is no default value.	application=ftpapp
partition	Specifies the partition this adapter handles as defined in iStudio. Any alphanumeric string is a possible value. There is no default value.	partition=germany
instance_number	Specifies the instance number to which this adapter corresponds. Specify a value only if you want to have multiple adapter instances for the given application with the given partition. Possible values are any integer greater than or equal to 1. There is no default value.	instance_number=1
agent_log_level	Specifies the amount of logging necessary. Possible values are: 0=errors only 1=status and errors 2=trace, status, and errors The default value is 1.	agent_log_level=2
agent_ subscriber_name	Specifies the subscriber name used when this adapter registers its subscription. The possible value is a valid Oracle Advanced Queue subscriber name. There is no default value.	agent_subscriber_ name=ftpapp
agent_message_ selector	Specifies conditions for message selection when registering its subscription with the hub. The possible value is a valid Oracle Advanced Queue message selector string. There is no default value.	agent_message_ selector=recipient_list like '%,aqapp,%'
agent_reply_ subscriber_name	Specifies the subscriber name used when multiple adapter instances for the given application with the given partition are used. This parameter is optional if there is only one instance running. The possible value is the application name (*parameter:application*) concatenated with the instance number (*parameter:instance_ number*). There is no default value.	If application=ftpapp and instance_number=2, then agent_reply_subscriber_ name=ftpapp2

Table 2–5   Agent Connection Parameters

Parameter	Description	Example
agent_reply_message_selector	Used only if there are multiple adapter instances for the given application with the given partition. The possible value is a string built using the concatenated application name (*parameter*:*application*) with the instance number (*parameter*:*instance_number*). There is no default value.	If application=ftpapp and instance_number=2, then agent_reply_message_selector=recipient_list like '%,ftpapp2,%'
agent_tracking_enabled	Specifies if message tracking is enabled. Set this parameter to false to turn off all tracking of messages. Set this parameter to true to track messages with tracking fields set in iStudio. Possible values are true or false. The default value is true.	agent_tracking_enabled=true
agent_throughput_measurement_enabled	Specifies if throughput measurement is enabled. Set this parameter to true to turn on all throughput measurements. Possible values are true or false. The default value is true.	agent_throughput_measurement_enabled=true
agent_use_custom_hub_dtd	Specifies whether to use a custom document type definition (DTD) for the common view message when handing it to the hub (the repository in which metadata is stored). By default, adapters use an Oracle9*i*AS InterConnect-specific DTD for all messages sent to the hub, as other Oracle9*i*AS InterConnect adapters retrieve the messages from the hub and know how to interpret them. Set this parameter to true if for every message, the DTD imported for the message of the common view is used instead of the Oracle9*i*AS InterConnect DTD. Only set this parameter to true if an Oracle9*i*AS InterConnect adapter is not receiving the messages from the hub. Possible values are true or false. There is no default value.	agent_use_custom_hub_dtd=false
agent_metadata_caching	Specifies the metadata caching algorithm. Possible values are: ■ startup—Cache everything at startup. This may take a while if there is a lot of metadata in the repository. ■ demand—Cache metadata as it is used. ■ none—No caching. This slows down performance. The default value is demand.	agent_metadata_caching=demand

Table 2–5   Agent Connection Parameters

Parameter	Description	Example
agent_dvm_table_ caching	Specifies the domain value mapping (DVM) table caching algorithm. Possible values are: ■ startup—Cache all DVM tables at startup. This may take a while if there are many tables in the repository. ■ demand—Cache tables as they are used. ■ none—No caching. This slows down performance. The default value is demand.	agent_dvm_table_ caching=demand
agent_lookup_ table_caching	Specifies the lookup table caching algorithm. Possible values are: ■ startup—Cache all lookup tables at startup. This may take a while if there are many tables in the repository. ■ demand—Cache tables as they are used. ■ none—No caching. This slows down performance. The default value is demand.	agent_lookup_table_ caching=demand
agent_delete_ file_cache_at_ startup	With any of the agent caching methods enabled, metadata from the repository is cached locally on the file system. Set this parameter to true to delete all cached metadata on startup. Possible values are true or false. The default value is false. **Note:** After changing metadata or DVM tables for this adapter in iStudio, you must delete the cache to guarantee access to the new metadata or table information.	agent_delete_file_cache_ at_startup=false
agent_max_ao_ cache_size	Specifies the maximum number of application objects' metadata to cache. Possible values are any integer greater than or equal to 1. The default value is 200.	agent_max_ao_cache_ size=200
agent_max_co_ cache_size	Specifies the maximum number of common objects' metadata to cache. Possible values are any integer greater than or equal to 1. The default value is 100.	agent_max_co_cache_ size=100

Table 2–5   Agent Connection Parameters

Parameter	Description	Example
agent_max_ message_ metadata_cache_ size	Specifies the maximum number of messages' metadata (publish/subscribe and invoke/implement) to cache. Possible values are any integer greater than or equal to 1. The default value is 200.	agent_max_message_ metadata_cache_size=200
agent_max_dvm_ table_cache_size	Specifies the maximum number of DVM tables to cache. Possible values are any integer greater than or equal to 1. The default value is 200.	agent_max_dvm_table_ cache_size=200
agent_max_ lookup_table_ cache_size	Specifies the maximum number of lookup tables to cache. Possible values are any integer greater than or equal to 1. The default value is 200.	agent_max_lookup_table_ cache_size=200
agent_max_queue_ size	Specifies the maximum size to which internal Oracle9*i*AS InterConnect message queues can grow. Possible values are any integer greater than or equal to 1. The default value is 1000.	agent_max_queue_size=1000
agent_ persistence_ queue_size	Specifies the maximum size to which internal Oracle9*i*AS InterConnect persistence queues can grow. Possible values are any integer greater than or equal to 1. The default value is 1000.	agent_persistence_queue_ size=1000
agent_ persistence_ cleanup_interval	Specifies how often to run the persistence cleaner thread (in milliseconds). Possible values are any integer greater than or equal to 30000. The default value is 60000.	agent_persistence_ cleanup_interval=60000
agent_ persistence_ retry_interval	Specifies how often for the persistence thread to retry when it fails to send an Oracle9*i*AS InterConnect message. Possible values are any integer greater than or equal to 60000. The default value is 60000.	agent_persistence_retry_ interval=60000
service_path	Windows only. Specifies the value to which to set the environment variable PATH. The PATH variable is set to the specified value before forking the Java VM. Typically, all directories containing all necessary DLLs should be listed here. Possible values are the valid path environment variable setting. There is no default value.	service_ path=%JREHOME%\bin;D:\ora cle\ora902\bin
service_ classpath	Specifies the class path used by the adapter Java VM. If a custom adapter is developed and, as a result, the adapter is to pick up any additional jars, add the jars to the existing set of jars being picked up. Possible values are the valid class path. There is no default value.	service_ classpath=D:\oracle\ ora902\oai\902\lib\ oai.jar;%JREHOME%\lib\rt. jar;D:\oracle\ora902\jdbc \classes12.zip

Table 2–5   Agent Connection Parameters

Parameter	Description	Example
service_class	Specifies the entry class for the Windows NT service. A possible value is `oracle/oai/agent/service/AgentService`. There is no default value.	`service_class=oracle/oai/agent/service/AgentService`
service_max_ java_stack_size	Windows only. Specifies the maximum size to which the Java VM's stack can grow. Possible values are the valid Java VM maximum native stack size. The default value is the default for the Java VM.	`service_max_java_stack_size=409600`
service_max_ native_stack_ size	Windows only. Specifies the maximum size to which the Java VM's native stack can grow. Possible values are the valid Java VM maximum native stack size. The default value is the default for the Java VM.	`service_max_native_stack_size=131072`
service_min_ heap_size	Windows only. Specifies the minimum heap size for the adapter Java VM. Possible values are the valid Java VM heap sizes. The default value is the default Java VM heap size.	`service_min_heap_size=536870912`
service_max_ heap_size	Windows only. Specifies the maximum heap size for the adapter Java VM. Possible values are any valid Java VM heap sizes. The default value is `536870912`.	`service_max_heap_size=536870912`
service_num_vm_ args	Windows only. Specifies the number of `service_vm_arg`*number* parameters specified. Possible values are the number of `service_vm_arg`*number* parameters. There is no default value.	`service_num_vm_args=1`

service_vm_ arg*number*	Windows only. Specifies any additional arguments to the Java VM. For example, to get line numbers in any of the stack traces, set `service_vm_arg1=java.compiler=NONE`. If there is a list of arguments to specify, use multiple parameters as shown in the example by incrementing the last digit starting with `1`. Be sure to set `service_num_vm_args` correctly. Possible values are any valid Java VM arguments. There is no default value.	`service_vm_arg1=java.compiler=NONE`
		
		`service_vm_arg2=oai.adapter=database`

FTP Adapter-Specific Parameters

Table 2–6 lists the parameters specific to the FTP adapter. With the exception of the `bridge_class` parameter, all parameters can be edited after installation.

Table 2–6   FTP Adapter-Specific Values

Parameter	Description	Example
bridge_class	Specifies the entry class for the FTP adapter. A value must be specified and cannot be modified later. A possible value is `oracle.oai.agent.adapter.technology.TechBridge`. There is no default value.	`bridge_class=oracle.oai.agent.adapter.technology.TechBridge`
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>`. The possible values are `ftp://<host name>/<directory path>`. There is no default value.	`ota.send.endpoint=ftp://ip-sun/private/ipdev1/test/inbound`
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>`. The possible values are `ftp://<host name>/<directory path>`. There is no default value.	`ota.receive.endpoint=ftp://ip-sun/private/ipdev1/test/inbound`
ota.type	Defines the type of payload this adapter handles. Possible values are XML and D3L. There is no default value.	`ota.type=XML`
ota.d3ls	Specifies the list of data definition description language (D3L) XML files used by this 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 SMTP adapter, the parameter must be updated and the SMTP adapter restarted. There is no default value.	`ota.d3ls=person.xml,person1.xml`
file.sender.user	The FTP user name for the outbound FTP server. The possible value is a valid FTP user name. There is no default value.	`file.sender.user=joe`
file.sender.type	Indicates the file types. The possible values are ASCII or BINARY. The default value is BINARY.	`file.sender.type=ASCII`

Table 2–6   FTP Adapter-Specific Values

Parameter	Description	Example
file.sender. proxy_host	The name of the machine that server as the proxy server for the outbound FTP server. The possible value is any correct host name. There is no default value.	file.sender.proxy_ host=www-proxy.foo.com
file.sender. proxy_port	The port number of the proxy server for the outbound FTP server. The possible value is any valid port number. There is no default value.	file.sender.proxy_ port=80
file.receiver. user	The FTP user name for the inbound FTP server. The possible value is any valid FTP user name. There is no default value.	file.receiver.user=joe
file.receiver. proxy_host	The name of the machine that servers as the proxy server for the inbound FTP server. The possible value is a correct host name. There is no default value.	file.receiver.proxy_ host=www-proxy.foo.com
file.receiver. proxy_port	The port number of the proxy server for the inbound FTP server. The possible value is a valid port number. There is no default value.	file.receiver.proxy_ port=80
file.receiver. exception_dir	The value of this parameter should be a URL which represents either an FTP directory or a file location. For this release, if an FTP URL can only be specified for the exception directory if the receiving endpoint is also an FTP URL, and furthermore the host name in the URL has to be 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. The user should make sure 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.	file.receiver. execption_ dir=ftp://acme.com/ private/user/error or file.receiver.exception_ dir=file://localhost/ private/user/error

3

Design Time and Runtime Concepts

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

- FTP Adapter Design Time Concepts
- FTP Adapter Runtime Concepts
- Starting the FTP Adapter
- Stopping the FTP Adapter
- FTP Error Code

FTP Adapter Design Time Concepts

The FTP adapter can handle XML and D3L structured payloads. For example:

- Pure XML data—string beginning with `<xml...`

- Fixed layout, typically binary data described by a D3L XML file.

> **See Also:** *Oracle9iAS InterConnect User Guide*

XML Payload

Users can import a DTD in iStudio which governs how the FTP adapter will parse a received XML document into a Oracle9iAS InterConnect application view event. In addition, the DTD describes how an inbound application view message will be converted into an XML document. Use the `XML` message type when defining a new integration point in any of the event wizards.

> **Note:** The `oracle.oai.agent.adapter.technology.type` parameter in the `adapter.ini` file should be set to `XML` instead of `D3L`.

When the adapter operates in the XML payload mode, no transformations are performed on the messages between, native view and application view, that are sent or received through the FTP adapter. This is apart from the implied straight ASCII <-> Java object conversion or parsing. Any XSLT transformations are performed before sending an XML document to Oracle9iAS InterConnect or after receiving one from Oracle9iAS InterConnect.

D3L Payload

The FTP adapter supports both XML and D3L data types. It converts and transforms application view messages to native format and back again.

When an application based on the FTP adapter is created in iStudio, the D3L message type and data type can be used. When these options are selected, messages received or sent by the FTP adapter must adhere to the fixed byte level layout defined in an D3L XML file.

When creating a common view in iStudio, the D3L import option can also be used.

> **See Also:** *Oracle9iAS InterConnect User Guide*

FTP Adapter Runtime Concepts

This section describes the key runtime components of the FTP adapter.

How the FTP Adapter Works

This section describes the following components of the FTP adapter:

- FTP Receiver—From remote FTP server or local file system to Oracle9*i*AS InterConnect functionality.

- FTP Sender—From Oracle9*i*AS InterConnect to remote FTP server functionality.

FTP Receiver

On the receiving side, 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>`

Based on header name and value information, the FTP bridge uses D3L/XML to parse from native-format into an Oracle9*i*AS InterConnect message object and generates an application view event. The agent converts the application view event into a common view event and hands it off to Oracle9*i*AS InterConnect for further routing and processing. Once the message is successfully handled off to Oracle9*i*AS InterConnect, the corresponding FTP file on the remote FTP server or local file system will be removed. If an error occurs, 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, the file will be discarded.

> **Note:** The value of the exception directory should be a URL which represents either an FTP directory or a file location. For this release, an FTP URL can only be specified for the exception directory if the receiving endpoint is also an FTP URL, and furthermore the host name in the URL has to be 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. The user should make sure 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.

The properties for the FTP receiver are found in the `adapter.ini` file as `file.receiver.*`.

FTP Sender

The FTP adapter supports sending to multiple FTP endpoints. This feature provides flexibility for sending messages to different remote FTP servers. An endpoint can be associated with a subscribing event in the iStudio by adding transport properties such as FTP endpoint, FTP user name, and password for this endpoint as metadata or modified fields for the event. Once the association of the endpoint and event is established, the message from the subscribing event is sent out to the FTP endpoint.

For example, the following metadata are associated to an event called SendOrder which sends an order to a FTP server at foo.com in the `/private/user/test/directory`.

Parameter	Description
`ota.endpoint=sendOrderAppEP`	Specifies a unique endpoint name.

`oracle.oai.agent.adapter.technology.`
`send.endpoint=ftp://foo.com/private/`
`user/test`
	Defines the FTP endpoint.

`file.sender.user=joe`	Defines the FTP user credentials.
`file.sender.password=welcome`	

See Also: *Oracle9iAS InterConnect User Guide*

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 multi-endpoint feature provides the flexibility to send messages to different FTP servers. The file name at the destination site is automatically generated by the adapter and is in the following form:

```
<ftp adapter name><instance number>-<time stamp>
```

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

Starting the FTP Adapter

Start the FTP adapter using the `start` script in the directory named after the FTP
adapter. On Windows or Windows 2000, start it from the Service window available
from the Start menu.

1. Access the Services window from the Start menu:

On...	Choose...
Windows NT	Start > Settings > Control Panel > Services
Windows 2000	Start > Settings > Control Panel > Administrative Tools > Services

The Services window displays.

2. Select the *OracleHome9iASInterConnectAdapter-Application* service.

3. Start the service based on your operating system:

On...	Choose...
Windows NT	Choose Start.
Windows 2000	Right click the service and choose Start from the menu that displays.

See Also: "FTP Adapter Configuration" on page 2-9for the
location of the `start` script

Sample Log File of Successfully Started Advanced Queuing Adapter

The following file displays an FTP adapter that was started successfully:

```
D:\oracle\ora902\oai\9.0.2\adapters\ftpapp>D:\oracle\ora902\oai\9.0.2•in\JavaSer
vice.exe -debug "Oracle OAI Adapter 9.0.2 - ftpapp"
D:\oracle\ora9021\oai\9.0.2\adapters\ftpapp adapter.ini
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.
```

Stopping the FTP Adapter

Stop the FTP adapter using the `stop` script in the directory named after the FTP adapter. On Windows NT or Windows 2000, stop the adapter from the Services window available from the Start menu.

1. Access the Services window from the Start menu:

On...	Choose...
Windows NT	Start > Settings > Control Panel > Services
Windows 2000	Start > Settings > Control Panel > Administrative Tools > Services

The Services window displays.

2. Select the *OracleHome9iASInterConnectAdapter-Application* service.

3. Start the service based on your operating system:

On...	Choose...
Windows NT	Choose Stop.
Windows 2000	Right click the service and choose Stop from the menu that displays.

Stop status can be verified by viewing the `oailog.txt` files in the appropriate timestamped subdirectory of the log directory of the adapter directory.

> **See Also:** "FTP Adapter Configuration" on page 2-9 for the location of the `start` script

FTP Error Code

This section defines the error codes that the FTP adapter returns in the event of an exception.

The error code returned by the FTP adapter corresponds to the negative acknowledgment sent by the remote FTP server to the FTP adapter. The full list of the FTP reply codes can be found in Section 4.2.2 of RFC 959.

The FTP reply code consists of three digits. There are five possible values for the first digit.

The following is the interpretation of the reply code for the negative acknowledgment.

```
4xx:   Transient Negative Completion reply
          The command is not accepted and the request action did not take place.
           The error condition is transient and can be retried.
5xx : Permanent Negative Completion reply
          The command was not accepted and the request action did not take
place.
           The error condition is permanent and the user should  not retried.
```

The second digit of the reply code corresponds to different functions:

```
x0x:  syntax error
x1x: Information
x2x: Connections
x3x: Authentication and accounting
x4x: Unspecified
x5x: File system
```

> **See Also:**   *Oracle9iAS InterConnect User Guide*

4

Frequently Asked Questions

This chapter provides answers to frequently asked questions about the FTP adapter. This chapter discusses the following topics:

- Installation Questions
- Design Time Questions

Installation Questions

The following questions address installation of the FTP adapter.

How do I know the FTP Adapter has started properly?

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

Platform	Directory
UNIX	`$ORACLE_HOME/oai/9.0.2/adapters/`*Application*`/log/`*timestamp_in_milliseconds*
Windows	`%ORACLE_HOME%\oai\9.0.2\adapters\`*Application*`\log\`*timestamp_in_milliseconds*

If there are no exceptions, 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 some idea about what went wrong. It is possible that the FTP adapter is unable to connect to the repository. Make sure 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:** *Oracle9iAS 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/OAI/9.0.2/adapters/`*Application*`/`
Windows	`%ORACLE_HOME%\oai\9.0.2\adapters\`*Application*`\`

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

Can I install multiple FTP adapters on the same machine?

The installer overwrites previous installations of the FTP adapter if it is installed a second time in the same Oracle home. However, multiple Oracle Homes can exist on one machine and have one FTP adapter is installed in each Oracle home. An Oracle home can be created by installing the Oracle8*i* client in different locations. When the FTP adapter is installed a second time, choose an Oracle home different from where the first FTP adapter is installed.

Design Time Questions

The following are design time questions for the FTP adapter.

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 the information from iStudio which is stored in the repository locally for better performance in a production environment. If you change something in iStudio and want to see the change in the runtime, you need to stop the FTP adapter which are effected, 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 any of the inifile parameters?

In order to encrypt any values specified in `.ini` file, complete the following steps:

1. Locate value to be encrypted.

2. Run the `encrypt` utility in `$OAI_HOME/bin` directory to encrypt above value.

3. Prefix the name of the parameter in the `.ini` file with `encrypted_`.

4. Replace the value with the new encrypted value from step 2.

Example: To encrypt the password for the `ftp.sender.password` parameter, replace `ftp.sender.password=ftpuser` with the following:

```
encrypted_ftp.sender.password=112411071071106510801094108410731070107110811069
```

A

Sample Adapter.ini File

This appendix shows a sample adapter.ini file for the FTP adapter. This appendix contains this topic:

■   adapter.ini Sample File

> **See Also:**   "FTP Adapter Configuration" on page A-2 for
> additional information on `adapter.ini` configuration parameters

Sample Adapter.ini File

The following code sample displays the FTP adapter `adapter.ini` file.

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

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

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

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

// If you want to have multiple Adapter instances for the given application with
the given part
ition, 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=


//-----------------------------------
// 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=

// 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 t
he 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
since other OA
I Adapters will be picking up the messages from the Hub and know how to
interpret them. This sh
ould be set to true if for every message, you would like to use the DTD imported
```

for that messa
ge'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 (cach
e 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 table
s 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), de
mand (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 de
mand) 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 star
ted, 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 data type information to cache

```
agent_max_ao_cache_size=200

// Max number of common data type 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

I