Oracle® Application Server Integration InterConnect

Adapter for SMTP Installation and User's Guide

10*g* Release 2 (10.1.2)

Part No. B14075-01

November 2004

ORACLE®

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

Part No. B14075-01

Contents

A   Frequently Asked Questions

B   Example of the adapter.ini File

Index

Send Us Your Comments

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

Part No. B14075-01

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 SMTP 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 SMTP (SMTP adapter), and the hardware and software requirements.

Chapter 2, "Installation and Configuration"

This chapter describes installation and configuration of the SMTP adapter.

Chapter 3, "Design Time and Runtime Concepts"

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

Appendix A, "Frequently Asked Questions"

This appendix provides answers to frequently asked questions about the SMTP 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 Text

We use the following conventions in text to help you more quickly identify special terms. The table also 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* 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. 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.
`lowercase monospace (fixed-width) font`	Lowercase monospace typeface indicates executable programs, filenames, directory names, and sample user-supplied elements. *Note:* Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to start 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. Connect as `oe` user. 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`. 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}` `[COMPRESS	NOCOMPRESS]`
...	Ellipsis points mean repetition in syntax descriptions.	`CREATE TABLE ... AS subquery;`
	In addition, ellipsis points can mean an omission in code examples or text.	`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);` `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. Because these terms are not case sensitive, you can use them in either UPPERCASE or lowercase.	`SELECT last_name, employee_id FROM`
		`employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

lowercase	Lowercase typeface indicates user-defined programmatic elements, such as 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;`

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 Oracle10*g*, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory.	Change to the *ORACLE_BASE**ORACLE_HOME*\rdbms\admin directory.
	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*.	
	All directory path examples in this guide follow OFA conventions.	
	Refer to *Oracle Database Installation Guide for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	

1

Introduction

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

- SMTP Adapter Overview
- SMTP Adapter System Requirements
- Known SMTP Adapter Limitations

1.1 SMTP Adapter Overview

The SMTP adapter enables an Oracle SMTP application to be integrated with other applications using OracleAS Integration InterConnect. The SMTP adapter is useful in all Enterprise Application Integration (EAI) environments where e-mail uses the Internet Message Access Protocol 4 (IMAP4) and SMTP. EAI is the integration of applications and business processes within the same company.

The SMTP adapter can monitor incoming messages in the form of e-mail placed on an IMAP server. The SMTP adapter is also capable of sending messages to SMTP servers. The payload type (the type of data being delivered to a destination) for this adapter is one of the following:

- Extensible Markup Language (XML) data
- Data definition description language (D3L) data

Figure 1–1 depicts the data flow of incoming messages from an IMAP server to OracleAS Integration InterConnect.

Figure 1–1   Incoming Messages
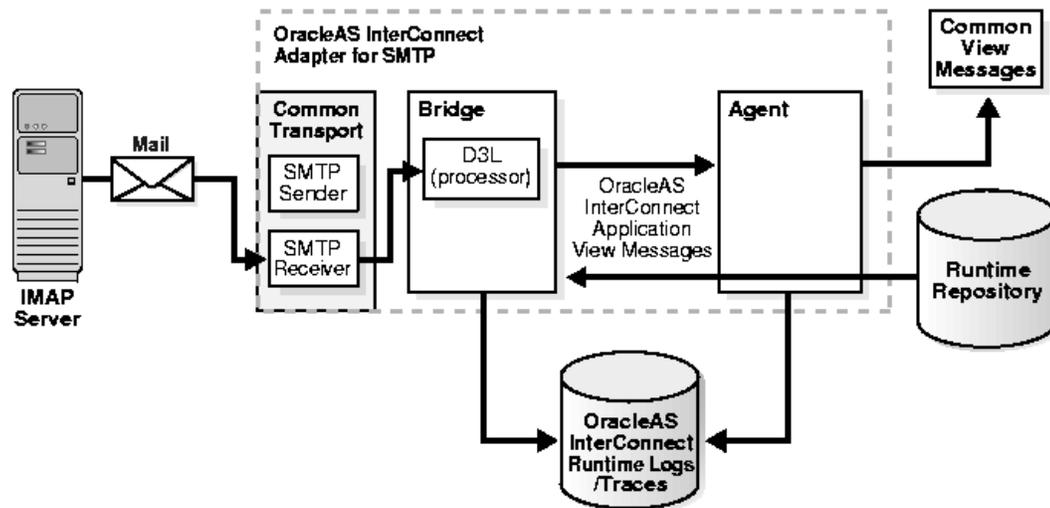
Figure 1–2 depicts the data flow of outgoing messages from OracleAS Integration InterConnect to an SMTP server.

Figure 1–2   Outgoing Messages
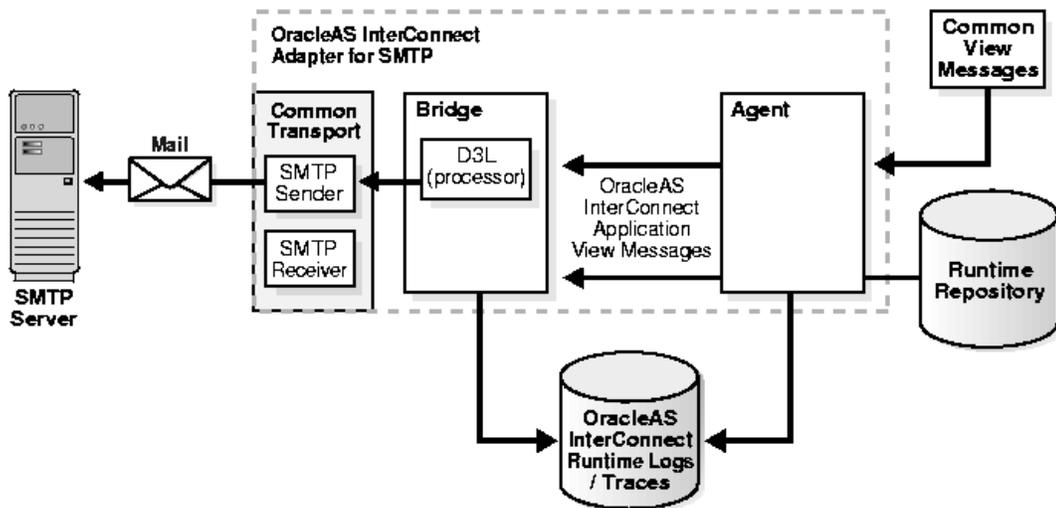
1.2 SMTP Adapter System Requirements

The following sections describe the system requirements for the SMTP adapter:

- Hardware Requirements
- Software Requirements

1.2.1 Hardware Requirements

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

Table 1–1 Hardware Requirements

Hardware	Windows 2000	UNIX
Disk space	500 MB	500 MB
Memory	128 MB	128 MB

1.2.2 Software Requirements

The following sections describe the software requirements for the SMTP adapter:

- Operating System Requirements
- JRE Requirements

Operating System Requirements

Table 1–2 lists the operating system requirements for the computer where the SMTP 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
	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 SMTP Adapter Limitations

The SMTP adapter has the following limitations:

- The IMAP server with Secure Socket Layer (SSL) is not supported.
- Only a single endpoint is supported for incoming messages.
- The sending and receiving applications must support SMTP.
- Only IMAP4 server is supported in this release. Post Office Protocol 3 (POP3) is not supported.
- In multiple instance cases, ensure that each application instance has a separate receiving endpoint. Otherwise, different instances of adapters may attempt to process the same message concurrently.
- For messages of type D3L, the message must be one part Multipurpose Internet Mail Extension (MIME), with the data encoded in base64. If the incoming email contains more than one attachment, then only the first attachment is extracted as payload.

2

Installation and Configuration

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

- Installing the SMTP Adapter
- Configuring the SMTP Adapter

2.1 Installing the SMTP Adapter

The SMTP adapter must be installed in an existing Oracle home Middle Tier for OracleAS 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 SMTP 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 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 SMTP adapter installation.

2.1.2 Installation Tasks

To install the SMTP adapter:

1. In the Available Product Components page of the OracleAS Integration InterConnect installation, select **SMTP adapter**, and click **Next**.

2. The Set Oracle Wallet Password screen is displayed. Enter and confirm the password on the screen, which will be used to administer OracleAS Integration InterConnect installation. Click **Next**.

- Go to step 3, if installing the SMTP 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 SMTP 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.

3. The Specify Hub Database Connection page 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 SMTP Adapter Name page is displayed.

5. Enter the application to be defined. Blank spaces are not permitted. The default value is `mySMTPApp`.

6. Click **Next**. The Specify SMTP Adapter Usage screen is displayed.

7. Select one of the 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:

- Email Address: The e-mail address of the outgoing SMTP server to which OracleAS Integration InterConnect sends messages. Enter the e-mail address as follows:

  `username@hostname`

- Outgoing Mail Server: The hostname of the outgoing SMTP server to which OracleAS Integration InterConnect sends messages

9. Click **Next**. The installation screen that appears is based on the selection made in Step 7:

If You Selected...	Then Go to Step...
Configure for both sending and receiving messages	10
Configure for sending messages ONLY	12

10. Enter the following information in the Configure Receiving Endpoint Information page:

   - Username: The user name account of the IMAP server from which the OracleAS Integration InterConnect receives messages.

   - Password: The password for the user name account.

   - Incoming Mail Server: The hostname of the IMAP server from which OracleAS Integration InterConnect receives messages.This information is required for polling the user name account and sending information back to OracleAS Integration InterConnect.

   Caution: For testing purposes, do not specify a personal e-mail account as the receiving endpoint. During runtime, the SMTP adapter connects to the IMAP server and removes messages in the folder specified by the receiving endpoint. Oracle recommends you create a dedicated e-mail account for testing and deploying this adapter.

   ---

11. Click **Next**. The Summary page is displayed.

12. Select **Install** to install the SMTP adapter. The adapter is installed in the following directory:

Platform	Directory
Windows	`ORACLE_HOME\intergration\interconnect\adapters\Application`
UNIX	`ORACLE_HOME/integration/interconnect/adapters/Application`

   `Application` is the value specified in Step 5.

13. Click **Exit** on the Installation page to exit the SMTP adapter installation.

2.1.3 Postinstallation Tasks

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

- Customizing the Payload Type

- Customizing the Sending Endpoints

- Customizing the Receiving Endpoints

2.1.3.1 Customizing the Payload Type

Payload data is the data sent between applications. To change the payload type from the default of XML to D3L, edit the parameters in the `adapter.ini` file.

To customize the payload data type:

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 SMTP 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 SMTP application. For example:

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

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

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. Set the `smtp.sender.content_type` parameter to the message content type to use. For example:

```
smtp.sender.content_type=plain/text
```

2. Set the `smtp.sender.character_set` parameter to the message character set to use. For example:

```
smtp.sender.character_set=iso-2022-jp
```

> **See Also:** Table 2–9, " SMTP Adapter-Specific Parameters"

2.1.3.3 Customizing the Receiving Endpoints

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

1. Set the `smtp.receiver.exception_folder` parameter to the folder name that can store files. For example:

```
smtp.receiver.exception_folder=error_messages
```

This parameter is not automatically set to a default value during installation. The IMAP administrator must create this folder. Leave this setting blank if you do not want to save these files.

2. Set the `smtp.receiver.polling_interval` parameter to the time interval in milliseconds to poll the IMAP server for messages. This parameter automatically defaults to a value of `10000` during installation. For example:

```
smtp.receiver.polling_interval=20000
```

3. Set the `smtp.receiver.max_msgs_retrieved` parameter to the maximum number of messages to retrieve in a polling session. This parameter automatically defaults to a value of `10` during installation. For example:

```
smtp.receiver.max_msgs_retrieved=30
```

> **See Also:** Table 2–9, " SMTP Adapter-Specific Parameters"

2.2 Configuring the SMTP Adapter

After an SMTP 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   SMTP Adapter Directory

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

Table 2–2 describes the various executable files of the SMTP adapter.

Table 2–2   SMTP Executable Files

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

> **See Also:**   "SMTP Adapter Error Codes" on page 3-10 for a list of error codes

Table 2–3 describes the SMTP adapter configuration files.

Table 2–3   SMTP Configuration Files

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

> **See Also:**   Appendix B, "Example of the adapter.ini File"

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

Table 2–4   SMTP Directories

Directory	Description
logs	The adapter activity is logged in subdirectories of the logs directory. Each new run of the adapter creates a subdirectory for the oailog.txt log file.
persistence	The messages are made available in this directory. Do not edit this directory or its files.

2.2.1 Ini File Settings

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

- hub.ini Files

- adapter.ini Files

2.2.1.1 hub.ini Files

The SMTP 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=InterConnectRepos itory`

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, then 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`

2.2.1.2 adapter.ini Files

The agent component of the SMTP adapter reads the `adapter.ini` file at runtime to access SMTP 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. Possible Value: Any valid port number. 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. Possible Values: `true` or `false`. Default Value: `false`. **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. Possible values: ■ `startup`: Cache all DVM tables at startup. This may take a while if there are a lot of tables in the repository. ■ `demand`: Cache tables as they are used. ■ `none`: No caching. This slows down performance. Default Value: `demand`.	agent_dvm_table_ caching=demand
agent_log_level	Specifies the amount of logging necessary. Possible values: 0=errors only 1=status and errors 2=trace, status, and errors Default Value: 1.	agent_log_level=2
agent_lookup_table_ caching	Specifies the lookup table caching algorithm. Possible values: ■ `startup`: Cache all lookup tables at startup. This may take a while if there are a lot of tables in the repository. ■ `demand`: Cache tables as they are used. ■ `none`: No caching. This slows down performance. Default Value: `demand`.	agent_lookup_table_ caching=demand
agent_max_ao_cache_ size	Specifies the maximum number of application object metadata to cache. Possible Value: An integer greater than or equal to `1`. Default Value: `200`.	agent_max_ao_cache_ size=200
agent_max_co_cache_ size	Specifies the maximum number of common object metadata to cache. Possible Value: An integer greater than or equal to `1`. Default Value: `100`.	agent_max_co_cache_ size=100

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

Parameter	Description	Example
agent_max_dvm_ table_cache_size	Specifies the maximum number of DVM tables to cache. Possible Value: An integer greater than or equal to 1. 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. Possible Value: An integer greater than or equal to 1. 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. Possible Value: An integer greater than or equal to 1. 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. Possible Value: Any integer greater than or equal to 1. 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. Possible Value: A valid Oracle Advanced Queue message selector string (like '%,aqapp,%'). Default Value: None.	`agent_message_ selector=%,aqapp,%`
agent_metadata_ caching	Specifies the metadata caching algorithm. Possible values: ■ `startup`: Cache everything at startup. This may take time if there are a lot of tables in the repository. ■ `demand`: Cache metadata as it is used. ■ `none`: No caching. This slows down performance. Default Value: demand.	`agent_metadata_ caching=demand`
agent_persistence_ cleanup_interval	Specifies how often to run the persistence cleaner thread in milliseconds. Possible Value: Any integer greater than or equal to 30000 milliseconds. Default Value: 60000.	`agent_persistence_ cleanup_ interval=60000`
agent_persistence_ queue_size	Specifies the maximum size of internal OracleAS Integration InterConnect persistence queues. Possible Value: Any integer greater than or equal to 1. 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. Possible Value: Any integer greater than or equal to 5000 milliseconds. Default Value: 60000.	`agent_persistence_ retry_interval=60000`

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

Parameter	Description	Example
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. Possible Value: `true`, `false` 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. Possible Value: `true`, `false`. 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 only if multiple adapter instances exist for the given application and given partition. Possible Value: A string built using the application name (parameter:application) concatenated with the instance number (parameter:instance_number). Default Value: None.	If `application=smtpapp`, `instance_number=2`, then `agent_reply_message_selector=receipient_list` like '`%,smtpapp2,%`'
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. Possible Value: The application name (parameter:application) concatenated with the instance number (parameter:instance_number). Default Value: None.	If `application=smtpapp` and `instance_number=2`, then `agent_reply_subscriber_name=smtpapp2`
agent_subscriber_name	Specifies the subscriber name used when this adapter registers its subscription. Possible Value: A valid Oracle Advanced Queue subscriber name. Default Value: None.	`agent_subscriber_name=smtpapp`
agent_throughput_measurement_enabled	Specifies if the throughput measurement is enabled. Set this parameter to `true` to turn on all throughput measurements. 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 all tracking of messages. Set this parameter to `true` to track messages with tracking fields set in iStudio. 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. 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. Default Value: None.	`agent_use_custom_hub_dtd=false`

Table 2–7 (Cont.) adapter.ini 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 while creating metadata. Possible Value: Any alphanumeric string. Default Value: None.	`application=smtpapp`
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. Possible Value: A valid character encoding. Default Value: `UTF-8`. 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`
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. Possible Value: A URL. 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 want to have multiple adapter instances for the given application with the given partition. Possible Value: Any integer greater than or equal to `1`. Default Value: None.	`instance_number=1`
nls_country	Specifies the ISO country code. The codes are defined by ISO-3166. 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 Default Value: `US`. **Note**: This parameter specifies date format. It is applicable for the date format only.	`nls_country=US`

nls_date_format	Specifies the format for date fields expressed as string.
	
	Possible Value: Any valid date format pattern as shown in Table 2–8 for the definitions of the format characters.
	
	Default Value: `EEE MMM dd HHmmss zzz yyyy`.
		Date format pattern `dd/MMM/yyyy` can represent 01/01/2003.
		
		`nls_date_format=dd-MMM-yy`
		
		Multiple date formats can be specified as `num_nls_formats=2`
		
		`nls_date_format1=dd-MMM-yy`
		
		`nls_date_format2=dd/MMM/yy`

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

Parameter	Description	Example
nls_language	Specifies the ISO language code. The codes are defined by ISO-639. 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 Default Value: en. **Note**: This parameter specifies date format. It is applicable for the date format only.	`nls_language=en`
partition	Specifies the partition this adapter handles as specified in iStudio. Possible Value: Any alphanumeric string. Default Value: None.	`partition=germany`
service_class	Specifies the entry class for the Windows service. Possible Value: `oracle/oai/agent/service/AgentService`. 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 jar files to the existing set of jar files. Possible Value: A valid `PATH` setting. Default Value: None. 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. Possible Value: A valid `jvm.dll`. Default Value: `jvm.dll`. This parameter is for Microsoft Windows only.	`service_jdk_dll=jvm.dll`
service_jdk_version	Specifies the JDK version that the adapter Java VM should use. Possible Value: A valid JDK version number. Default Value: 1.4 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. Possible Value: A valid JVM heap size. Default Value: `536870912`. This parameter is for Microsoft Windows only.	`service_max_heap_size=536870912`
service_max_java_stack_size	Specifies the maximum size that the JVM stack can grow. Possible Value: A valid JVM maximum stack size. Default Value: Default value for the JVM. This parameter is for Microsoft Windows only.	`service_max_java_stack_size=409600`

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

Parameter	Description	Example
service_max_native_ stack_size	Specifies the maximum size the JVM native stack can grow. Possible Value: The valid JVM maximum native stack size. Default Value: Default value for the JVM. 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. Possible Value: The valid JVM heap size. Default Value: `536870912`. This parameter is for Microsoft Windows only.	`service_min_heap_ size=536870912`
service_num_vm_args	Specifies the number of `service_vm_arg`*number* parameters specified in JVM. Possible Value: The number of `service_vm_ arg`*number* parameters. Default Value: None. This parameter is for Microsoft Windows only.	`service_num_vm_args=1`
service_path	Specifies the environment variable `PATH`. The `PATH` variable before starting the Java Virtual Machine (JVM). Typically, list all directories that contain necessary DLLs. Possible Value: The valid `PATH` environment variable setting. Default Value: None. 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 of the 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`.
	
	Possible Value: A valid JVM arguments.
	
	Default Value: None.
	
	This parameter is for Microsoft Windows only.
		`service_vm_ arg1=java.compiler= NONE`
		
		`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`

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

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

SMTP Adapter-specific Parameters

Table 2–9 lists the parameters specific to the SMTP adapter.

Table 2–9   SMTP Adapter-Specific Parameters

Parameter	Description	Example
bridge_class	Specifies the entry class for the SMTP adapter. A value must be specified and cannot be modified later. Possible Value: `oracle.oai.agent.adapter.technology.TechBridge`. Default Value: None..	`bridge_class=oracle.oai.agent.adapter.technology.TechBridge`
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. When 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. Default Value: None.	`ota.d3ls=person.xml, person1.xml`
ota.receive.endpoint	Specifies the receiving endpoint URL for the SMTP adapter. The URL is written as follows: `imap://username@imapHostName` Default Value: None.	`ota.receive.endpoint=imap://joe@server10`
ota.send.endpoint	Defines the sending endpoint URL for the SMTP adapter. The URL is written as follows: `mailto:username@hostname` Default Value: None.	`ota.send.endpoint=mailto:joe.one@test.com`
ota.time_format	Specifies the format in which you want to define the timestamp. The format options are identical to `java.text.SimpleDateFormat`. If you don't want to define the `smtp_sender_customizer_class` parameter, but still want to define the subject line generation, then use this parameter. Default Value: None.	`ota.time_format=yyyyMMddHHmmss`

Table 2–9   (Cont.) SMTP Adapter-Specific Parameters

Parameter	Description	Example
ota.time_format_couuntry_code	Specifies the country code. 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. Possible Value: A valid ISO country code as mentioned in `java.util.Locale`. Default Value: None.	`ota.time_format_couuntry_code=FR`
ota.time_format_language_code	Specifies the language code. 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. Possible Value: A valid ISO language code as mentioned in `java.util.Locale`. Default Value: None.	`ota.time_format_language_code=fr`
ota.type	Specifies the type of payload this adapter handles. 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 Possible Value: `XML` or `D3L`. Default Value: `XML`.	`ota.type=XML`
smtp.receiver.customizer_class	Specifies the class name for customization. Used by the SMTP receiver. Default Value: `oracle.oai.aget.adapter.technology.DefaultReceiverCustomizer`	`smtp.receiver.customizer_class=MySMTPSenderCustomizer`
smtp.receiver.exception_folder	Specifies a mail folder in which to put e-mails that cannot be processed successfully. This mail folder must be created by the IMAP server administrator. Possible Value: A valid mail folder name. Default Value: None.	`smtp.receiver.exception_folder=error`
smtp.receiver.max_msgs_retrieved	Specifies the maximum number of messages to be retrieved in each polling session. Possible Value: An integer greater than `0`. Default Value: `30`.	`smtp.receiver.max_msgs_retrieved=10`

Table 2–9   (Cont.) SMTP Adapter-Specific Parameters

Parameter	Description	Example
smtp.receiver.pass word	Specifies the user password for the IMAP server. This password can also be encrypted by running the encrypt tool and renaming this parameter to `encrypted_smtp.receiver.password`. Possible Value: A valid password string. Default Value: None. **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.	`smtp.receiver.password= smtpuser`
smtp.receiver.pollin g_interval	Specifies the time interval during which to poll the IMAP server (in milliseconds). Possible value: An integer greater than `0`. Default Value: `60000` (`60` seconds).	`smtp.receiver.polling_ interval=10000`
smtp.receiver.proto col	Specifies the e-mail protocol to use. For this release, the only possible value is `imap`. Default Value: None.	`smtp.receiver.protocol= imap`
smtp.sender.charact er_set	Specifies the character encoding for the message. Default Value: None.	`smtp.sender.character_ set=iso-2022-jp`
smtp.sender.content _type	Specifies the content type of e-mail messages (RFC 822 header field). Default Value: None.	`smtp.sender.content_ type=plain/text`
smtp.sender.custom izer_class	Specifies the class name for customization. Used by the SMTP sender. Default Value: `oracle.oai.agent.adapter.technology.SMTP DefaultSenderCustomizer`	`smtp.sender.customizer_ class=MySMTPSenderCustomi zer`
smtp.sender.smtp_ host	Specifies the SMTP host to use in sending messages. Default Value: None.	`smtp.sender.smtp_ host=smtp1.foo.com`
smtp.sender.subject _rule	Specifies the rule for generating subject. Used by the SMTP sender. Default Value: `%APP%%PART%_%TIME%`	`smtp.sender.subject_ rule=Message_from_%APP%_ %EVENT%_%TIME%`

3

Design Time and Runtime Concepts

This chapter describes the design time and runtime concepts for the Simple Mail Transfer Protocol (SMTP) adapter. It contains the following topics:

- SMTP Adapter Design Time Concepts
- SMTP Adapter Runtime Concepts
- Customizing the SMTP Adapter
- Starting the SMTP Adapter
- Stopping the SMTP Adapter
- SMTP Adapter Error Codes

3.1 SMTP Adapter Design Time Concepts

The SMTP 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

You can import a Document Type Definition (DTD) in iStudio to determine how the SMTP adapter parses a received XML document into an OracleAS 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 SMTP 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 SMTP adapter performs a two-way conversion and transformation of messages between application view and native format.

An application based on the SMTP adapter can use the iStudio Message Type D3L and the iStudio D3L Data Type Import options when importing a data type. In this case,

messages received or sent by the SMTP 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 and common view datatypes

3.2 SMTP Adapter Runtime Concepts

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

- SMTP Receiver
- SMTP Sender
- SMTP Adapter Message Format

3.2.1 SMTP Receiver

The SMTP adapter receives incoming messages from a single receiving endpoint, which is an e-mail address on an Internet Message Access Protocol (IMAP) server.

The endpoint is of the form: `imap://username@imapHostName`

During each polling interval, the SMTP receiver:

- Polls the IMAP server for incoming e-mails
- Processes each e-mail
- Transforms the e-mail message into a transport message processed by the SMTP bridge. You can configure the maximum number of e-mails processed for each session through the `smtp.receiver.max_msgs_retrieved` parameter of the `adapter.ini` file.

The polling interval is configured using the smtp.receiver.polling_interval parameter of the adapter.ini file.

The SMTP bridge uses the D3L XML file based on name/value pairs or magic value message header attributes (a sequence of bytes in the native format message header). The SMTP bridge uses this information to parse from the native message to an OracleAS InterConnect message object and translate it to an application view event. The agent converts the application view event to a common view event and passes it to OracleAS InterConnect for further routing and processing.

Once the message is successfully passed to OracleAS Integration InterConnect, the corresponding e-mail residing on the IMAP server is marked to be deleted, and is deleted at the end of each session. An exception folder on the IMAP server can be specified for storing the unsuccessfully processed e-mails. The exception folder can be set using the `smtp.receiver.exception_folder` parameter in the `adapter.ini` file. If no exception folder is set, then the mail is deleted.

The properties for the SMTP receiver are defined in the `adapter.ini` file and take the form of `smtp.receiver.*`.

See Also:

- *Oracle Application Server Integration InterConnect User's Guide*, Appendix B, for additional information on D3L name-value pair and magic value message header attributes

- Figure 1–1, "Incoming Messages" on page 1-2

- "SMTP Adapter-specific Parameters" on page 2-13

3.2.2 SMTP Sender

The SMTP adapter consists of the SMTP 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 XML to perform the conversion of the common view object to the native format. The native format message is then sent through the SMTP transport layer to an SMTP endpoint.

The SMTP adapter's sending endpoint is written as follows:

```
mailto:username@hostname
```

The subject header of each message sent by the SMTP adapter is automatically generated by the adapter and is in the following syntax:

```
SMTP_adapter_application_namepartition-time_stamp
```

You can specify a rule for generating the subject when the SMTP adapter sends an email. To use this feature, add the parameter, `smtp.sender.subject_rule`, in 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,

```
smtp.sender.subject_rule=Message_from_%APP%_%EVENT%_%TIME%
```

This rule instructs the SMTP adapter to generate subject with the following pattern:

```
Message_from_your app name_event name_current time stamp
```

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

An endpoint is associated with a subscribing event in iStudio by adding the transport properties for the SMTP endpoint 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 SMTP endpoint.

When using the multiple endpoint feature with `XML` data type, you must use the `Generic` event type, 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 with an event called `sendOrder` that sends messages to an e-mail account `scott@tiger.com`.

Table 3–1   SendOrder Event Metadata

Parameter	Description
`ota.endpoint=sendOrderAppEP`	Specifies a unique endpoint name set in iStudio
`ota.send.endpoint=mailto:scott@ tiger.com`	Specifies the sending endpoint for the SMTP adapter

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

If no metadata is associated with an event in iStudio, then the endpoint specified by the `ota.send.endpoint` parameter in the `adapter.ini` file is used as the default endpoint.

The properties for the SMTP sender are defined in the `adapter.ini` file and take the form of `smtp.sender.*`.

> **See Also:**
>
> - Figure 1–2, "Outgoing Messages" on page 1-2
> - "SMTP Adapter-specific Parameters" on page 2-13
> - Chapter 4 of the *Oracle Application Server Integration InterConnect User's Guide* for information on adding transport properties as metadata in iStudio

3.2.3 SMTP Adapter Message Format

This section describes how to extract and send messages to the SMTP adapter for different types of payloads.

If the SMTP adapter operates in D3L mode, then the message format is binary or plain text. The message must be sent or received as one part Multipurpose Internet Mail Extension (MIME), with the data encoded in base64. Example 3–1 shows how to send the message to the SMTP adapter in MIME format using the `JavaMail` API.

Example 3–1   Sending Messages to the SMTP Adapter

```
Message smtpMessage = new MimeMessage(session);
String msg = new String("This is a test.");
MimeBodyPart part = new MimeBodyPart();

// create a multipart object
Multipart mp = new MimeMultipart();
DataSource dataSource = new BytesDataSource(msg.getBytes());
part.setDataHandler(new DataHandler(dataSource));
part.setHeader("Content-Transfer-Encoding", "base64");
mp.addBodyPart(part);
smtpMessage.setContent(mp);
...
Transport.send(smtpMessage);
```

In Example 3–1, `BytesDataSource` is a user-written class that implements the `DataSource` class, which represents a data source consisting of a byte array. Refer to the `JavaMail` API for additional information.

Example 3–2 shows how to extract the multipart message sent from the SMTP adapter when it operates in D3L mode.

Example 3–2   Extracting Messages Sent from the SMTP Adapter

```
Object o  = message.getContent();
Multipart mp = (Multipart)o;

// The message is contained in the
// first part.
BodyPart part = mp.getBodyPart(0);
InputStream is = (InputStream)part.getContent();

// extract the data from input stream.
...
```

When the SMTP adapter operates in XML mode, the message is sent or received in simple text format, as described in RFC 822. To send a message to the SMTP adapter, use the `javax.mail.Message.setText()` method in the `JavaMail` API.

3.3 Customizing the SMTP Adapter

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

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

- `oracle.oai.adapter.agent.technology.SMTPSenderCustomizer`

3.3.1 ReceiverCustomizer Interface

You can use the `ReceiverCustomizer` interface to customize the `TransportMessage` object that is received by the SMTP adapter. 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 describes 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
`customizeTransportMessage();`	Allows you to customize the transport message received by the adapter. It uses the following parameters:
	agent: Log a message.
	receiverType: Information on the type of adapter.
	transportMessage: Customize the transport message received by the adapter.
`createReplyMessage();`	Creates a reply message based on the status and the message received. It contains the following parameters:
	agent: Log a message.
	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 not processed successfully.
	receivedTransportMessage: Transport message received by the adapter. This parameter is used to transport headers in the transport message to create a meaningful message.
	The return string contains the reply message. This method is used for backward compatibility for the SMTP adapter. However, for the SMTP adapter, you should return a `null` value with this method.

Example 3–3   Example of ReceiverCustomizer Interface

The `MyReceiverCustomizer` class to remove 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 from an email
// that OracleAS Integration InterConnect does not understand.
   public void customizeTransportMessage(Agent agent, int receiverType,
                                         TransportMessage transportMessage)
    {
       String payload = transportMessage.getBodyAsString();
// For debugging purposes only, the following syntax removes the first line
// from the payload. Details of removeFirstLine() is not provided.
       agent.logTraceMessage("payload received = " + payload, null, null, null);
       String newPayload = removeFirstLine(payload);

       try {
          transportMessage.setBody(newPayload);
       }
       catch(TransportException te) {
       . . . .
       }
    }
// For the SMTP adapter, a null string from the following method will be returned.
   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 use for the
`TransportMessage` class.

Method	Description
public String toString();	Dump messages and headers.
public void setTransportHeader(String name, String value);	Set a transport-specific header.
public Properties getTransportHeaders();	Get all transport-specific headers and return a Properties object that contains all the transport headers.
public void setBody(String body) throws TransportException;	Set the body of the message. The body type will be set to STRING. Parameter includes:
	body: body of the message
	It throws a TransportException.
public void setBody(InputStream in) throws TransportException;	Set the body of the message. The body type will be set to BYTES. Parameter includes:
	InputStream: Contains the message.
	It throws a TransportException.
public String getBodyAsString();	Get the body of the message as String object. Return the message in String object.
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.

3.3.2 SMTPSenderCustomizer Interface

You can use the `SMTPSenderCustomizer` interface to customize the subject name
and payload of the `TransportMessage` object that is sent to the transport layer. The
`SMTPSenderCustomizer` interface extends the `SenderCustomizer` interface. You
must implement the `SMTPSenderCustomizer` interface by implementing the
following two methods:

- `SMTPSenderCustomizer.customizeTransportMessage()`

- `SMTPSenderCustomizer.generateSubjectName()`

If you do not want to implement the `generateSubjectName() method`, then you
can create a class that extends the
`oracle.oai.agent.adapter.technology.SMTPDefaultSenderCustomizer`
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 SMTPSenderCustomizer Interface

The following code describes the file structure of the `SMTPSenderCustomizer` 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 SMTPSenderCustomizer extends SenderCustomizer {
      public String generateSubjectName(Agent agent,
                                        String rule,
                                        String app,
                                        String partition,
                                        Date time,
                                        MessageObject mobj,
                                        AttributeObject aobj);

   }
```

generateSubjectName method

This method generates an subject name for email. It contains the following parameters:

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

rule: Rule for generating subject. This parameter is read from `smtp.sender.subjectRule` 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 subject name.

3.4 Starting the SMTP Adapter

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

- To start the SMTP 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 SMTP 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 SMTP adapter using the IC Manager. Refer to *OracleAS Integration InterConnect User's Guide* for more details.

3.4.1 Log File of SMTP Adapter

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

```
timestamp_in_milliseconds
```

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

```
The Adapter service is starting..
Registering your application (SMTPAPP)..
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 SMTP Adapter

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

- To stop the SMTP 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**.

- On Windows, stop the SMTP adapter from Services.

    1. Access the Services window from the Start menu.

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

    The Services window is displayed.

    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 SMTP adapter by viewing the `oailog.txt` files. These files are located in the timestamped subdirectory of the `log` directory of the SMTP adapter.

3.6 SMTP Adapter Error Codes

This section defines the error codes (derived from the `JavaMail` exception) that the SMTP adapter returns in the event of an exception.

```
OTA-IMAP-1002
Reason: Authentication failed due to bad user name or password.
Action: Check user name or password.

OTA-IMAP-1003
Reason: Folder closed exception is thrown when a method is invoked on
an invalid Messaging Object or Folder Object.
Action: None.

OTA-IMAP-1004
Reason: Message removed exception. A method is invoked on an expunge message.
Action: None.

OTA-IMAP-1005
```

```
Reason: Read-only folder exception. Tried to write to a read-only folder.
Action: Check the properties of the folder. Make sure it has the correct write
privilege.

OTA-SMTP-1001
Reason: Message cannot be sent exception.
Action: Make sure the email address for sending is valid.
```

> **See Also:** *Oracle Application Server Integration InterConnect User's Guide* for information on the retry action

A

Frequently Asked Questions

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

- How do I know the SMTP adapter started properly?
- The SMTP adapter did not start properly: what is wrong?
- Is it possible to edit the SMTP adapter configuration settings created during installation?
- When I change an element in iStudio, such as mappings, it seems like the SMTP adapter uses old information. What is happening?
- Can I install multiple SMTP adapters on the same computer?
- How do I secure my passwords?

How do I know the SMTP adapter started properly?

View the `oailog.txt` file located in the timestamped subdirectory of the SMTP adapter `logs` directory.

On...	Change to...
UNIX	*ORACLE_ HOME*/integration/interconnect/adapters/*Application*/logs/*ti mestamp_in_milliseconds*
Windows	*ORACLE_ HOME*\integration\interconnect\adapters*Application*\logs*ti mestamp_in_milliseconds*

The SMTP adapter did not start properly: what is wrong?

View the exceptions in the adapter log file (`oailog.txt`).

The exceptions should provide information about what went wrong. It is possible that the SMTP adapter is unable to connect to the repository. Ensure the repository is started properly. The SMTP adapter will connect to the repository once it is started properly. You do not need to restart the Adapter.

> **See Also:** *Oracle Application Server Installation Guide* for instructions on starting the repository on UNIX and Windows

Is it possible to edit the SMTP 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*\

> **Note:** All configuration parameters with the exception of `bridge_class` can be edited more than once.

> **See Also:**

When I change an element in iStudio, such as mappings, it seems like the SMTP adapter uses old information. What is happening?

The SMTP 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 SMTP adapter, delete the SMTP adapter cache files, and restart the SMTP adapter.

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

Can I install multiple SMTP adapters on the same computer?

The installer overwrites previous installations of the SMTP adapter if you try to install it a second time in the same Oracle home. However, you can have multiple Oracle homes on a computer and have one SMTP adapter in each Oracle home. When you install the SMTP adapter a second time, choose a different Oracle home from the first SMTP adapter.

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 an `adapter.ini` example file for the SMTP adapter.

> **See Also:** "Configuring the SMTP Adapter" on page 2-5 for additional information on `adapter.ini` configuration parameters

This section shows an `adapter.ini` example file for the SMTP adapter.

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

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

// Application (as created in iStudio) that this Adapter corresponds to.
application=smtpapp1

// 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 partition, each Adapter should have an instance number.

//instance_number=2

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

// define the type of payload. Valid option is XML or D3L.
ota.type=D3L

// define the smtp sending endpoint
ota.send.endpoint=mailto:ipdev2@cc-sun.us.oracle.com

// define the smtp receiving endpoint
ota.receive.endpoint=imap://joe@server10

//-------------------------------------
// SMTP Sender initialization variables
//-------------------------------------

// specify the smtp host
smtp.sender.smtp_host=smtp1.foo.com

// Specify the content type for the email
// smtp.sender.content_type=plain/text
```

```
// Specify the character set for the email
// smtp.sender.character_set=iso-2022-jp

//---------------------------------------
// SMTP Receiver initialization variables
//---------------------------------------

// enter the email user's password
encrypted_smtp.receiver.password=112411071071106510801094108410731070107110811069

// enter the email server protocol.
smtp.receiver.protocol=imap

// email folder name
smtp.receiver.folder=inbox

// polling interval in milli seconds
smtp.receiver.polling_interval=60000

// maximum number of messages that
// the receiver will retrieve for
// each polling session (default 30)
smtp.receiver.max_msgs_retrieved = 10

// exception folder for messages
// that are not processed successfully.
smtp.receiver.exception_folder= error

// A list of the D3L XML files used by this Bridge. Each business event handled
// by the Bridge must have it's own D3L XML file.
// Whenever a new D3L XML file has been imported in iStudio to be used by
// an application using the SMTP adapter, the following parameter must
// be updated and the adapter restarted.
ota.d3ls=person.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=smtpapp1
agent_message_selector=recipient_list like '%,smtpapp1,%'
// 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
// because 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 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_dvm_table_caching=demand

// Sets the lookup table 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_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 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
```

Index

T

X