Oracle® Application Server InterConnect

Adapter for SMTP Installation and User's Guide

10*g* (9.0.4)

Part No. B10414-01

September 2003

ORACLE®

Oracle Application Server InterConnect Adapter for SMTP Installation and User's Guide, 10*g* (9.0.4)

Part No. B10414-01

Contents

3   Design Time and Runtime Concepts

4   Frequently Asked Questions

A   adapter.ini Example File

Index

Send Us Your Comments

Oracle Application Server InterConnect Adapter for SMTP Installation and User's Guide, 10g (9.0.4)

Part No. B10414-01

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: 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 (optionally) electronic mail address.

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

Preface

This guide is the primary source of introduction, installation, configuration, and usage information for the Simple Mail Transfer Protocol (SMTP) adapter.

This preface contains these topics:

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

Audience

Oracle Application Server InterConnect Adapter for SMTP Installation and User's Guide is intended for developers who want to integrate an SMTP application with other applications using Oracle Application Server 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 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.

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

Organization

This document contains:

Chapter 1, "Introduction"
This chapter describes the Oracle Application Server 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 for the SMTP adapter.

Chapter 4, "Frequently Asked Questions"
This chapter provides answers to frequently asked questions about the SMTP adapter.

Appendix A, "adapter.ini Example File"
This appendix shows an `adapter.ini` example file.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Application Server InterConnect User's Guide*

- *Oracle Application Server InterConnect Installation Guide*

- *Oracle Application Server InterConnect Adapter Configuration Editor User's 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://otn.oracle.com/membership/
```

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/documentation/
```

Conventions

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

- Conventions in Text
- Conventions in Code Examples
- Conventions for Microsoft Windows Operating Systems

Conventions in Text

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

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index**-**organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*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 italic monospace (fixed-width) font`	Lowercase italic monospace 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;`

Conventions for Microsoft Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Oracle Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > 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 file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	`c:\winnt"\"system32` is the same as `C:\WINNT\SYSTEM32`
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>`

	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 scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"`
		
		`C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)`

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 OracleHOME_NAMETNSListener`

1

Introduction

This chapter describes the Simple Mail Transfer Protocol (SMTP) adapter and the hardware and software requirements.

This chapter contains these topics:

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

SMTP Adapter Overview

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

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 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 InterConnect).
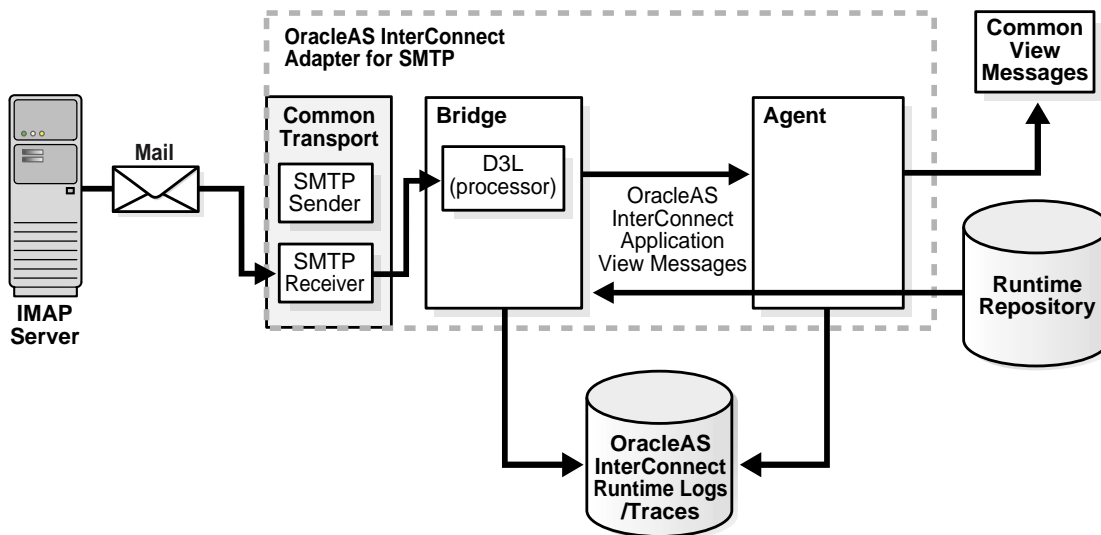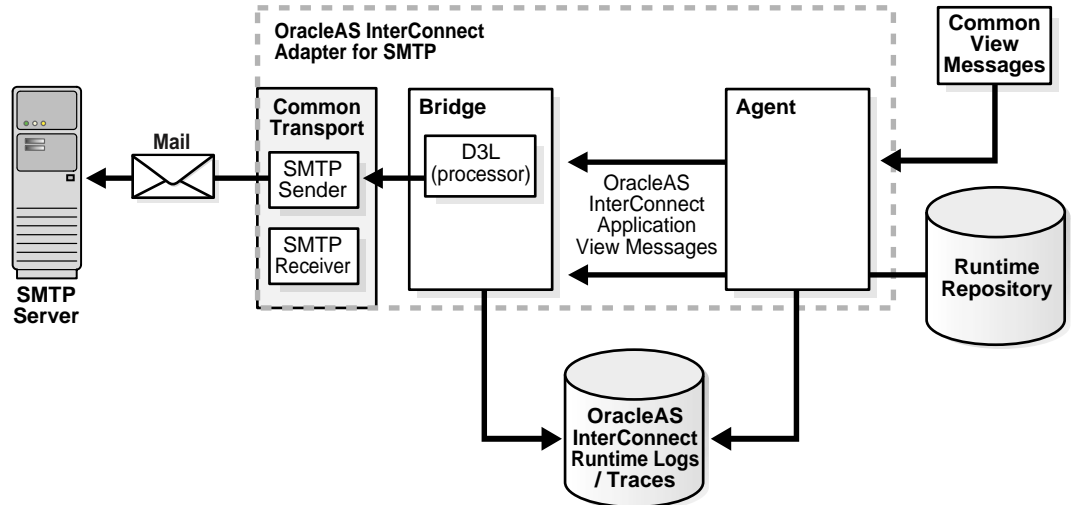
Figure 1–1   Incoming Messages

Figure 1–2 depicts the data flow of outgoing messages (from OracleAS InterConnect to an SMTP server).

Figure 1–2 Outgoing Messages

SMTP Adapter System Requirements

The following sections describe SMTP adapter system requirements:

- Hardware Requirements
- Software Requirements

Hardware Requirements

Table 1–1 lists the hardware requirements for the computer on which to install the SMTP adapter.

Table 1–1   Hardware Requirements

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

Software Requirements

The following sections describe SMTP adapter software requirements:

- Operating System Requirements
- JRE Requirements

Operating System Requirements

Table 1–2 lists the operating system requirements for the computer on which to install the SMTP adapter.

Table 1–2   Operating System Requirements

Operating System	Version
Windows NT	Version 4.0 with Service Pack 6 or above
Windows 2000	With Service Pack 1 or above
IBM AIX 5L	5.1 and 5.2 (64 bit)
HP Tru64	5.1a and 5.1b (64 bit)
HP-UX	11.0 and 11.11 (64 bit)
Red Hat Linux	Advanced Server 2.1
Sun SPARC Solaris	8 and 9 (32 bit)

JRE Requirements

OracleAS InterConnect uses Java Runtime Environment (JRE) 1.4.1, which is installed with its components.

SMTP Adapter Known Limitations

The SMTP adapter has the following limitations:

- The IMAP server with Secure Socket Layer (SSL) is not supported.

- Only multiple endpoints (destinations) for outgoing messages (from OracleAS InterConnect to SMTP servers) are supported. For incoming messages (IMAP server to OracleAS InterConnect), only a single endpoint is supported.

- The sending endpoint and receiving endpoints are restricted to SMTP endpoints.

- 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 SMTP adapter expects the message to be sent or received as a one part Multipurpose Internet Mail Extension (MIME) message with the data encoded in base64. If the incoming email contains more than one attachment, only the first attachment will be extracted as the payload.

2

Installation and Configuration

This chapter describes Simple Mail Transfer Protocol (SMTP) adapter installation and configuration.

This chapter contains these topics:

- Installing SMTP Adapter
- SMTP Adapter Configuration Parameters

Installing SMTP Adapter

This section contains these topics:

- Preinstallation Tasks
- Installation Tasks
- Postinstallation Tasks

Preinstallation Tasks

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

- An existing OracleAS InterConnect home Oracle home for 10*g* (9.0.4).
- A new Oracle home (Oracle Universal Installer creates this Oracle home for you)

Consult the following guides before proceeding with SMTP adapter installation:

- *Oracle Application Server Installation Guide,* which includes information on:
  - Oracle Universal Installer startup
- *Oracle Application Server InterConnect Installation Guide*, which includes information on:
  - CD-ROM mounting
  - OracleAS InterConnect software, hardware, and system requirements
  - OracleAS InterConnect installation

Note: OracleAS InterConnect Hub is installable through the OracleAS InterConnect Hub installation type. You must install the OracleAS InterConnect Hub before proceeding with the SMTP adapter installation.

Installation Tasks

To install the SMTP adapter:

1. On the Available Product Components page of the OracleAS InterConnect installation, select SMTP adapter, then select **Next**.

   Consider the following scenarios:

   - If installing the Oracle9*i* Database Server in an independent Oracle home, make sure that the OracleAS InterConnect Hub has been installed, not necessarily in the same Oracle home. Continue to step 2.

   - If installing the SMTP adapter in an existing Oracle home, make sure that it is a home directory to one of the OracleAS InterConnect components. Continue to step 3.

   > **Note:** The hub database information, such as the SID, host, port, and username/password from the Hub installation, is needed for step 2.

2. If installing OracleAS InterConnect for the first time on this machine, complete the following steps to enter the hub database information:

   a. On the Welcome page, select **Next**. The Database Configuration page displays. Enter information in the following fields:

      * Host Name—The host name of the machine where the hub database is installed.

      * Port Number—The TNS listener port for the hub database.

      * Database SID—The SID for the hub database.

   b. Click **Next**. The Database User Configuration page displays. Enter information in the following fields:

      * User Name—The hub database user name. Make sure the OracleAS InterConnect Hub is installed. If the Hub is not installed, complete the installation and note the user name and password.

      * Password—The password for the hub database user.

3. Click **Next**. The Adapter Configuration page displays. Enter the application to be defined or already defined in iStudio in the Application Name field. White spaces or blank spaces are not permitted. The default value is `mySMTPApp`.

4. Click **Next**. The Oracle Application Server InterConnect for SMTP Adapter usage page displays.

5. Select one of the following options and go to the step specified to enable the sending and receiving of messages, the sending of messages only, or the receiving of messages only from an external data source, such as an SMTP server. You can change the values for these 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	6
Configure for sending messages ONLY	6
Configure for receiving messages ONLY	8

6. Enter the following information in the OracleAS InterConnect SMTP Adapter Configuration - Configure sending endpoint information screen:

   ■ Email Address—The e-mail address of the outgoing SMTP server to which Oracle Application Server InterConnect sends messages. Enter the e-mail address as follows:

   username@hostname

   ■ Outgoing Mail Server—The hostname of the outgoing SMTP server to which Oracle Application Server InterConnect sends messages

7. Click **Next**.

   The installation screen that appears next is based on the selection you made in Step 5:

If You Selected...	Then Go to Step...
Configure for both sending and receiving messages	8
Configure for sending messages ONLY	10

8. Enter the following information in the OracleAS InterConnect SMTP Adapter Configuration - Configure receiving endpoint information screen:

   - Username—The username account of the IMAP server from which the Oracle Application Server InterConnect receives messages

   - Password—The password for the username account

   - Incoming Mail Server—The hostname of the IMAP server from which Oracle Application Server InterConnect receives messages. This information is required for polling the username account and sending information back to Oracle Application Server 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 Corporation recommends that you create a dedicated e-mail account for testing and deploying this adapter.

   ---

9. Click **Next**.

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

    When finished, the Summary screen appears.

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

Platform	Directory
Windows	*ORACLE_HOME*\oai\9.0.4\adapters*Application*
UNIX	*ORACLE_HOME*/oai/9.0.4/adapters/*Application*

You have defined the value of `Application` in Step 3.

Postinstallation Tasks

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

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

> **See Also:**
> - Table 2–1 on page 2-8 for the location of the `adapter.ini` file
> - Table 2–9 on page 2-19 for `adapter.ini` file parameter setting information specific to the SMTP adapter

Customizing the Payload Datatype

Payload data is the data sent between applications. If you want to change the payload datatype from the default of XML to the data definition description language (D3L), edit the following parameters in the `adapter.ini` file.

To customize the payload datatype:

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:** The following parameter descriptions for additional information:
   > - ota.d3ls on page 2-19
   > - ota.type on page 2-19

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.

To customize the sending endpoint:

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:** The following parameter descriptions for additional information:
>
> - `smtp.sender.content_type` on page 2-19
> - `smtp.sender.character_set` on page 2-19

Customizing the Receiving Endpoints

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

To customize the receiving endpoint:

1. Set the `smtp.receiver.exception_folder` to the folder name in which to place files that have not been processed successfully. 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 during which 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:** The following parameter descriptions for additional information:
>
> - `smtp.receiver.exception_folder` on page 2-20
> - `smtp.receiver.polling_interval` on page 2-20
> - `smtp.receiver.max_msgs_retrieved` on page 2-20

SMTP Adapter Configuration Parameters

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

Table 2–1   SMTP Adapter Directory

Platform	Directory
UNIX	*ORACLE_HOME*/oai/9.0.4/adapters/Application
Windows	*ORACLE_HOME*\oai\9.0.4\adapters\Application

Table 2–2   SMTP 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:** "SMTP Adapter Error Codes" on page 3-16 for a list of error codes

Table 2–3   SMTP Configuration Files

File	Description
ErrorCodes.ini (Windows and UNIX)	Contains 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, "adapter.ini Example File"

Table 2–4   SMTP Directories

Directory	Description
persistence	The messages are persisted (made available) in this directory. Do not edit this directory or its files.
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.

Hub.ini Parameters

The SMTP adapter connects to the hub database using parameters from the hub.ini file located in the hub directory. Table 2–5 lists the parameter name, description, the possible and default values, and example for each parameter

Table 2–5   Hub.ini Parameters

Parameter	Description	Example
hub_username	The name of the hub database schema (or username). The possible value is a valid hub database username. There is no default value.	hub_username=myhub
hub_password	The password for the hub database user. The possible value is a the valid password for the hub database user. There is no default value.	hub_password=manager
hub_host	The name of the machine hosting the hub database. The possible value is a the valid machine name. There is no default value.	hub_host=mpjoshipc

Table 2–5   Hub.ini Parameters

Parameter	Description	Example
hub_instance	The valid SID of the hub database. The possible value is a valid SID. There is no default value.	hub_instance=orcl
hub_port	The TNS listener port number for the HUB database instance. The possible value is a TNS listener port number. There is no default value.	hub_port=1521
repository_name	The valid name of the repository this adapter talks to. The possible value is a valid repository name. The default value is InterConnectRepository.	repository_name=InterConnectRepository

RAC-specific Hub.ini Parameters

When a hub is installed on a Real Application Cluster (RAC) database, parameters listed in Table 2–6 represent information on additional nodes used for connection and configuration. These parameters are added on top of the default parameters which represent the primary node. In Table 2–6, x represent the node number, which varies between 2 and the number of nodes. For example, if the RAC setup contains 4 nodes, x can take a value between 2 and 4.

Table 2–6   RAC-specific Hub.ini Parameters

Parameter	Description	Example
hub_num_nodes	Number of nodes in RAC cluster.	hub_num_nodes=4
hub_hostx	This parameter represents the host where the RAC database is installed.	hub_host2=dsunram13
hub_instancex	This parameter represents the instance on the respective node.	hub_instance2=orcl2
hub_portx	This parameter represents the port where the node has its instance available.	hub_port2=1521

Adapter.ini Parameters

This section contains these topics:

- Agent Connection Parameters
- SMTP Adapter-Specific Parameters

Agent Connection Parameters

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

Table 2–7   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=smtpapp
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=smtpapp
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,%'

Table 2–7   Agent Connection Parameters

Parameter	Description	Example

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 only one instance is 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=smtpapp` and `instance_number=2`, then `agent_reply_`
		`subscriber_`
		`name=smtpapp2`

agent_reply_
message_selector
	Used only if multiple adapter instances exist 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=smtpapp` and `instance_number=2`, then `agent_reply_`
		`message_`
		`selector=recipient_`
		`list like '%,smtpapp2,%'`

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 OracleAS InterConnect-specific DTD for all messages sent to the hub, as other OracleAS 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 OracleAS InterConnect DTD. Only set this parameter to `true` if an OracleAS 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`

Table 2–7   Agent Connection Parameters

Parameter	Description	Example
agent_metadata_ caching	Specifies the metadata caching algorithm. Possible values are: ■ startup—Cache everything at startup. This may take a while if there are a lot of tables 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
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 many tables are 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 many tables are 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

Table 2–7   Agent Connection Parameters

Parameter	Description	Example
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
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 OracleAS 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 OracleAS 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 milliseconds. The default value is 60000.	agent_persistence_ cleanup_interval=60000
agent_ persistence_ retry_interval	Specifies how often the persistence thread retries when it fails to send an OracleAS InterConnect message. Possible values are any integer greater than or equal to 60000 milliseconds. The default value is 60000.	agent_persistence_ retry_interval=60000
agent_pipeline_ to_hub	Specifies how to turn on or off the pipeline for messages from the Bridge towards the hub. If you set the pipeline to false, the file persistence is not used in that direction.	agent_pipeline_to_ hub=false
agent_pipeline_ from_hub	Specifies how to turn on or off the pipeline for messages from the hub towards the Bridge. If you set the pipeline to false, the file persistence is not used in that direction.	agent_pipeline_from_ hub=false

Table 2–7 Agent Connection Parameters

Parameter	Description	Example
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, list all directories here that contain all necessary DLLs. Possible values are the valid PATH environment variable setting. There is no default value.	service_ path=%JREHOME%\bin;D:\ oracle\ora904\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\ ora904\oai\904\lib\ oai.jar;%JREHOME%\lib\ rt.jar;D:\oracle\ora90 4\jdbc\classes12.zip
service_class	Specifies the entry class for the Windows 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

Table 2–7   Agent Connection Parameters

Parameter	Description	Example
service_vm_ argnumber	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 a list of arguments to specify exists, 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=NON E service_vm_ arg2=oai.adapter=datab ase
corba_port_ number	The CORBA port number on which the adapter CORBA service listens. Generally, this port is allocated dynamically. However, it can be configured to enable access across firewall.	corba_port_ number=14000
encoding	Character encoding for published messages. The adapter uses this parameter to generate encoding information in encoding tag of transformed OracleAS InterConnect message. OracleAS InterConnect represents messages internally as an XML document. The default encoding of the XML document is UTF-8. However, this encoding can be configured using this parameter, which is typically used when the OracleAS InterConnect message consists of characters not supported by UTF-8 and when the XMLParser is unable to handle them.	encoding=JA16SJIS

Table 2–7   Agent Connection Parameters

Parameter	Description	Example
nls_date_format	Format for date fields expressed as string. The default date format is `EEE MMM dd HH:mm:ss zzz yyyy`. For the meaning of this string, see the list of reserved characters in Table 2–8.	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`
nls_country	This parameter is a valid ISO Country Code. These upper-case and two-letter codes are defined by ISO-3166. You can find a full list of these codes at a Web site, such as, `http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html`	US
	The default Country code is `US`.	
	Note: This parameter specifies date format. It is applicable for the date format only.	
nls_language	This parameter is a valid ISO Language Code. These lower-case and two-letter codes are defined by ISO-639. You can find a full list of these codes at a Web site, such as, `http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt`	`nls_language=en`
	The default language code is `en`.	
	Note: This parameter specifies date format. It is applicable for the date format only.	

Table 2–8 shows the reserved characters used to specify the value of the nls_date_
format parameter. Using these characters, you can construct a pattern to define
date formats.

Table 2–8   Reserved Characters for the Value of the nls_date_format Parameter

Letter	Description	Example
G	Era designator	AD
y	Year	1996; 96
M	Month in year	July; Jul; 07
w	Week in year	27
W	Week in month	2
D	Day in year	189
d	Day in month	10
F	Day of week in month	Number 2
E	Day in week	Tuesday; 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
G	Era designator	AD

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. There is no default value. A possible value is `oracle.oai.agent.adapter.technology.TechBridge`.	`bridge_class=oracle.oai.agent.adapter.technology.TechBridge`
ota.receive.endpoint	Specifies the receiving endpoint URL for the SMTP adapter. There is no default value. The URL is of the form: `imap://username@imapHostName`	`ota.receive.endpoint=imap://joe@server10`
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`
ota.type	Specifies the message type the SMTP adapter handles for both incoming and outgoing messages. The options are `XML` or `D3L`. The default value is `XML`.	`ota.type=XML`
ota.send.endpoint	Specifies the sending endpoint URL for the SMTP adapter. There is no default value. The URL is of the form: `mailto:username@hostname`	`ota.send.endpoint=mailto:joe.one@test.com`
smtp.sender.smtp_host	Specifies the SMTP host to use in sending messages.	`smtp.sender.smtp_host=smtp1.foo.com`
smtp.sender.content_type	Specifies the content type of e-mail messages (RFC 822 header field).	`smtp.sender.content_type=plain/text`
smtp.sender.character_set	Specifies the character encoding for the message.	`smtp.sender.character_set=iso-2022-jp`
smtp.sender.subject_rule	Specifies the rule for generating subject. Used by the SMTP sender. The default value is: `%APP%%PART%_%TIME%`	`smtp.sender.subject_rule=Message_from_%APP%_%EVENT%_%TIME%`

Table 2–9   SMTP Adapter-Specific Parameters

Parameter	Description	Example
`smtp.sender.cus tomizer_class`	Specifies the class name for customization. Used by the SMTP sender. The default value is: `oracle.oai.agent.adapter.technology.SMTP DefaultSenderCustomizer`	`smtp.sender.customizer_ class=MySMTPSenderCustom izer`
`smtp.receiver.p assword`	User password for the IMAP server. The possible value is a valid password. There is no default value. This password can also be encrypted by running the encrypt tool and renaming this parameter to `encrypted_smtp.receiver.password`. **See Also:** "How do I make the adapter.ini file password parameter secure?" on page 4-3 for instructions on encrypting the user password	`smtp.receiver.password= smtpuser`
`smtp.receiver.p rotocol`	Specifies the e-mail protocol to use. For this release, the only possible value is `imap`. There is no default value.	`smtp.receiver.protocol= imap`
`smtp.receiver.e xception_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 values are a valid mail folder name. There is no default value.	`smtp.receiver.exception_ folder=error`
`smtp.receiver.p olling_interval`	Specifies the time interval during which to poll the IMAP server (in milliseconds). Possible values are any integers greater than `0`. The default value is `60000` (60 seconds).	`smtp.receiver.polling_ interval=10000`
`smtp.receiver.m ax_msgs_ retrieved`	Specifies the maximum number of messages to be retrieved in each polling session. Possible values are any integers greater than `0`. The default value is 30.	`smtp.receiver.max_msgs_ retrieved=10`
`smtp.receiver.c ustomizer_class`	Specifies the class name for customization. Used by the SMTP receiver. The default value is: `oracle.oai.aget.adapter.technology.Defau ltReceiverCustomizer`	`smtp.receiver.customizer _ class=MySMTPSenderCustom izer`

3

Design Time and Runtime Concepts

This chapter describes the design time and runtime concepts for the Simple Mail Transfer Protocol (SMTP) adapter.

This chapter contains these 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

SMTP Adapter Design Time Concepts

The SMTP adapter can handle XML and the data definition description language (D3L) structured payload data. For example:

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

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

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

XML Payload

You can import a document type definition (DTD) in iStudio that determines how the SMTP adapter parses a received XML document into an Oracle Application Server InterConnect application view event. In addition, the DTD describes how an inbound application view message is converted into an XML document. Use the message type XML when defining a new integration point in any of the Event Wizards.

You must also ensure that the `ota.type` parameter in the `adapter.ini` file is set to `XML` instead of `D3L`. When the adapter operates in XML payload mode, no translations are performed between native view and application view messages sent or received through the SMTP adapter. This is apart from the implied straight ASCII to Java object conversion (parsing). Any Extensible Stylesheet Language Transformations (XSLT) should be performed before sending or receiving an XML document to or from Oracle Application Server InterConnect.

D3L Payload

The SMTP adapter supports both XML and D3L datatypes. The SMTP adapter translates application view messages to native format and vice versa.

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

If preferred, the D3L Data Type Import option can also define common view datatypes.

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

SMTP Adapter Runtime Concepts

This section describes the two main SMTP adapter components.

This section contains these topics:

- SMTP Receiver
- SMTP Sender
- SMTP Adapter Message Format

> **See Also:** *Oracle Application Server InterConnect User's Guide*,
> Appendix B, for an example involving an Oracle adapter, the
> Advanced Queuing adapter, and D3L

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,
to Oracle Application Server InterConnect.

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

During each polling interval (configurable with the `adapter.ini` file
`smtp.receiver.polling_interval` parameter), the SMTP receiver performs
the following tasks:

- 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. The maximum number of e-mails processed for each session is user
  configurable with the `adapter.ini` file `smtp.receiver.max_msgs_`
  `retrieved` parameter.

Once the SMTP bridge detects a message, in the case of D3L payload, it uses a D3L
definition selected based on the name-value pair or magic value message header
attributes (a sequence of bytes in the native message header) to parse from native
format to an Oracle Application Server InterConnect message object and generates
an application view event. The agent transforms the application view event into a
common view event and sends it to Oracle Application Server InterConnect for
further routing and processing.

Once the message is successfully sent to Oracle Application Server InterConnect, the corresponding e-mail residing on the IMAP server is marked to be deleted, and is deleted at the end of each session. In the event that an error occurs, the IMAP server administrator can specify an exception folder on the IMAP server for storing the unsuccessfully processed e-mails with the `adapter.ini` file `smtp.receiver.exception_folder` parameter. If no exception folder is set, 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 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 2-19

SMTP Sender

The SMTP adapter supports sending outgoing messages from Oracle Application Server InterConnect to multiple SMTP endpoints. This feature provides flexibility for sending messages to different remote SMTP servers. An endpoint is associated with a subscribing event in iStudio by adding the transport properties for this endpoint as metadata. This is done through the Modify Fields button of the Subscribe Wizard - Define Application View dialog for the event. Once the association of the endpoint and event is established, the message from the subscribing event is sent to the SMTP endpoint.

For example, the metadata in Table 3–1 is associated with an event called `sendOrder`, which sends messages to an e-mail account `mailto: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 SMTP adapter's sending endpoint

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

The SMTP adapter is comprised 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 common view object to native format. The native format message is then sent through the SMTP transport layer to an SMTP endpoint.

The SMTP adapter's sending endpoint takes the following form:

`mailto:`*`username@hostname`*

The multiple endpoint feature enables messages to be sent to different SMTP servers. The subject header of each message sent by the SMTP adapter is automatically generated in the following form:

`SMTP_adapter_application_namepertition-time_stamp`

> **Note:** When using the multiple-endpoint feature with `XML` data type, you must choose the event type of `Generic`, instead of `XML`. Using the `Generic` event type allows you to enter the metadata for the endpoints via the Modify Fields feature associated with iStudio.

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

The user can specify a rule for generating the file name when the SMTP adapter sends an email. To use this feature, the user has to add the parameter, `smtp.sender.subject_rule`, in the `adapter.ini` file. The adapter recognizes the following tokens:

- `%APP%`—application name
- `%PART%`—partition number
- `%BO%`—business object name
- `%TYPE%`—message type
- `%EVENT%`—corresponding event name

%TIME%—time stamp

%MV%—message version

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>
```

If the above rule does not serve your needs, you can write your own customization rule by implementing the following interface:

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

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 1-3
> - "SMTP Adapter-Specific Parameters" on page 2-19
> - Chapter 4 of the *Oracle Application Server InterConnect User's Guide* for information on adding transport properties as metadata in iStudio

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 (the `ota.type` parameter is set to `D3L` in the `adapter.ini` file), the message format is binary or plain text. The SMTP adapter expects the message to be sent or received as a one part Multipurpose Internet Mail Extension (MIME) message 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. See 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 `ota.type` parameter is set to `XML` in the `adapter.ini` file), 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.

Customizing the SMTP Adapter

You can modify some of the SMTP adapter behaviors by implementing the following two interfaces:

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

ReceiverCustomizer Interface

You can use the `ReceiverCustomizer` interface to customize the `TransportMessage` object that is received by the SMTP adapter. The `customizeTransportMessage()` method can be used to customize the `TransportMessage` object before the adapter processes it. The `TransportMessage` object represents the native message that the transport layer receives or sends.

File Structure

The following is the file structure of this 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);
}
```

File Summary

The following table summarizes the `ReceiverCustomizer` Interface.

Methods	Description
`customizeTransportMes sage();`	This method allows you to customize the transport message, `Message`, received by the adapter. It contains the following parameters:
	`agent`—Used to log a message.
	`receiverType`—Provides information on the type of adapter.
	`transportMessage`—Used to customize the transport message received by the adapter.

Methods	Description
createReplyMessage();	This method creates a reply message, Message, based on the status and the message received. This method is used for backward compatibility. It contains the following parameters:
	agent—Used to log a message.
	status—The status of the message process. If the value is TransportResponse.TRANSPORT_ACK, the message is processed successfully. If the value is TransportResponse.TRANSPORT_ERROR, the message is processed unsuccessfully.
	receivedTransportMessage—The transport message is received by the adapter. This parameter is used to transport headers in the transport message to create a meaningful HTTP message.
	The return string contains the reply message. However, this method is used by the HTTP adapter. For the SMTP adapter, you should return a null value with this method.

Example 3–3   Example of ReceiverCustomizer Interface

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

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

public class MyReceiverCustomizer implements ReceiverCustomizer {
```

This example describes how to remove an extra line from an email that OracleAS InterConnect does not understand.

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

> **Note:**   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) {
. . . .
}
}
```

> **Note:** 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;
}
}
```

Example 3–4   List of Methods for the TransportMessage Class

This example provides a list of methods you may choose for the `TransportMessage` class.

Methods	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`.

Methods	Description
`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.

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()`

However, if you do not want to implement the more complicated `generateSubjectName()` method, 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.

SenderCustomizer Interface

File Structure
The following is 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);

}
```

File Summary

The following table summarizes the `customizeTransportMessage` method.

Method	Description
`customizeTransportMessage ();`	This method specifies how to customize the transport message for transporting sender. The adapter creates a TransportMessage for the transport layer to send based on the `MessageObject` sent by OracleAS InterConnect. You can use this method to further customize the transport message that is to be sent out by the transport layer.
	This method contains the following parameters:
	`agent`—Can be used to log message.
	`transportMessage`—Indicates the `TransportMessage` object that the adapter has created for sending.
	`mobj`—Indicates the `MessageObject` from OracleAS InterConnect.
	`aobj`—Indicates the `AttributeObject` from OracleAS InterConnect.
	This method does not return anything. You can change the payload with the `transportMessage` parameter.

SMTPSenderCustomizer Interface

File Structure

The following is 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;
```

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

import oracle.oai.agent.adapter.sdk.MessageObject;
import oracle.oai.agent.adapter.sdk.AttributeObject;
import java.util.Date;
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);

   }
```

File Summary

The following table summarizes the generateSubjectName method.

Method	Description
generateSubject Name ();	This method generates an subject name for email. It contains the following parameters:
	agent—Indicates that the user can use the Agent object to log message.
	rule—Indicates Rule for generating subject. This is read from smtp.sender.subjectRule in adapter.ini.
	app—Indicates the application name.
	partition—Indicates Partition.
	time—This is a Date object which indicates the time the object is received.
	mobj—Indicates a MessageObject passed from OracleAS InterConnect.
	aobj—Indicates an AttributeObject passed from OracleAS InterConnect.
	This method returns a string representing the file name.

Starting the SMTP Adapter

On UNIX, start the SMTP adapter using the `start` script located in the following directory:

`ORACLE_HOME/oai/9.0.4/adapters/Application`

 Type **start**, then press **Enter**.

On Windows, start the SMTP 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 appears.

2. Select the *OracleHomeOracleASInterConnectAdapter-Application* service.

3. Start the service based on the operating system:

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

> **See Also:** "SMTP Adapter Configuration Parameters" on page 2-8 for the location of the `start` script

Log File Example of Successfully Started SMTP Adapter

You can verify the startup status by viewing the `oailog.txt` files. These files are located in the appropriate timestamped subdirectory of the `log` directory of the SMTP adapter directory. Subdirectory names take the following form:

`timestamp_in_milliseconds`

The following file displays information about an SMTP adapter that started successfully:

```
D:\oracle\ora904\oai\9.0.4\adapters\smtpapp>D:\oracle\ora904\oai\9.0.4\bin\JavaS
ervice.exe -debug "Oracle OAI Adapter 9.0.4 -
smtpapp" D:\oracle\ora904\oai\9.0.4\adapters\smtpapp adapter.ini
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.
```

Stopping the SMTP Adapter

On UNIX, stop the SMTP adapter using the stop script located in the following directory named after the Oracle HTTP application.

ORACLE_HOME/oai/9.0.4/adapters/*Application*

Type **stop**, then press **Enter**.

On Windows, stop the SMTP 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 appears.

2. Select the *OracleHomeOracleASInterConnectAdapter-Application* service.

3. Stop the service based on the operating system:

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

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

> **See Also:** "SMTP Adapter Configuration Parameters" on page 2-8 for the location of the `stop` script

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 InterConnect User's Guide* for information on the retry action

4

Frequently Asked Questions

This chapter provides answers to frequently asked questions about the Simple Mail Transfer Protocol (SMTP) adapter.

SMTP Adapter Troubleshooting Questions

The following questions address troubleshooting issues for the SMTP adapter.

How do I know the SMTP adapter started properly?

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

On...	Go to...
UNIX	*ORACLE_HOME*/oai/9.0.4/adapters/*Application*/logs/*timestamp_in_milliseconds*
Windows	*ORACLE_HOME*\oai\9.0.4\adapters*Application*\logs*timestamp_in_milliseconds*

where `Application` is the value you defined in Step 3 on page 2-3 and `timestamp_in_milliseconds` is the directory. If no exceptions are listed, the adapter started properly.

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

View the exceptions in the adapter log file (`oailog.txt`). The exceptions provide information about inconsistencies. One possible reason is that the SMTP adapter did not connect to the repository. Make sure the repository is started properly and the SMTP adapter connects 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

I changed an element in iStudio, but the SMTP adapter uses old information - what is happening?

The SMTP adapter caches the information from iStudio (the information that is stored in the repository) locally for better performance in a production environment. If you change information in iStudio and want to see the change in the runtime environment, you must perform the following tasks:

To see iStudio changes in the runtime environment:

1. Stop the affected adapters.

2. Delete the adapter cache files.

3. Restart the adapter.

Each adapter has a `persistence` directory located in the directory named after the Oracle SMTP application. Deleting this directory when the adapter has been stopped makes the adapter obtain the new metadata from the repository when started.

If I cannot answer some SMTP configuration questions or I make a mistake during installation, can I edit these settings later?

Yes, edit the parameters in the following file:

On...	Go to...
UNIX	*ORACLE_HOME*/oai/9.0.4/adapters/*Application*/adapter.ini
Windows	*ORACLE_HOME*\oai\9.0.4\adapters*Application*\adapter.ini

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

> **See Also:** "Hub.ini Parameters" on page 2-9 for parameter information

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 make the adapter.ini file password parameter secure?

In order to encrypt password values specified in the `adapter.ini` file, perform the following steps:

To encrypt password values:

1. Locate the password value to encrypt.

2. Run the encrypt utility to encrypt the password value. The encrypt utility is located in the *ORACLE_HOME*/oai/9.0.4/bin directory for UNIX and the *ORACLE_HOME*\oai\9.0.4\bin directory for Windows. For example, to encrypt the `smtp.receiver.password` parameter, enter the following:

```
encrypt password
```

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

    ```
    encrypted_smtp.receiver.password
    ```

4. Replace the value with the new encrypted value created in Step 2. For example, to encrypt the password for the parameter `smtp.receiver.password`, replace the line

    ```
    smtp.receiver.password=smtpuser
    ```

    with the value you received from running the encrypt tool in Step 2. For example:

    ```
    encrypted_smtp.receiver.password=11241107107110651080109410841073107010711081069
    ```

A

adapter.ini Example File

This appendix shows an `adapter.ini` example file for the Simple Mail Transfer Protocol (SMTP) adapter.

This appendix contains this topic:

■ adapter.ini Example File

> **See Also:** "SMTP Adapter Configuration Parameters" on page 2-8 for additional information on `adapter.ini` configuration parameters

adapter.ini Example File

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

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

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

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

            // If metadata caching, DVM table caching, or lookup table caching are turned on
            (startup or demand) then the Adapter
```

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

// Max number of application 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