Oracle® One-to-One Fulfillment

Implementation Guide

Release 11*i*

Part No. B10634-02

August 2004

ORACLE®

Oracle One-to-One Fulfillment Implementation Guide, Release 11*i*

Part No. B10634-02

Primary Authors: Michael Phelan

Contributors: Prasad Kodur, Jacqueline Tessmer, Simon Lau, Shankar Radhakrishnan, Kwok Tung Law, Lenny Kluz, Sushant Agarwal, Sirisha Gandham, Kevin Craig Wong, Aravind Dongara, Sonali Vijay Inamdar, Edward Wang, Ravi Jayaraman, Girish Vasvani, Robert Wang, Rosie Wu, Victor TY Ho.

Contents

3 Before You Begin

4 Implementation Tasks

5 Integration Tasks

6 Verifying the Implementation

7 Diagnostics and Troubleshooting

8 Administration Tasks

Send Us Your Comments

Oracle One-to-One Fulfillment Implementation Guide, Release 11*i*

Part No. B10634-02

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 at:

Oracle Corporation
Oracle One-to-One Fulfillment Documentation
1900 Oracle Way
Reston, VA 20190
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

Intended Audience

Welcome to Release 11*i* of the Oracle One-to-One Fulfillment Implementation Guide.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Oracle One-to-One Fulfillment

  If you have never used Oracle One-to-One Fulfillment, Oracle suggests you attend one or more of the Oracle One-to-One Fulfillment training classes available through Oracle University.
- The Oracle Applications graphical user interface.

  To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*.

See Other Information Sources for more information about Oracle Applications product information.

How To Use This Guide

This document contains the information you need to understand and use Oracle One-to-One Fulfillment.

Chapter 1 describes the Interaction Center product family. It also describes what is new in the current release.

Chapter 2 provides a detailed description of this product.

Chapter 3 describes the system requirements and product dependencies. It also describes how to access the administrative console.

Chapter 4 provides a high-level overview of the implementation task sequence and references to procedures that are performed during the course of an implementation.

Chapter 5 explains the procedures that are necessary for integration with other Oracle applications.

Chapter 6 gives instructions on verifying that the implementation was done correctly.

Chapter 7 gives instructions on troubleshooting.

Chapter 8 describes task-based procedures for administering Oracle One-to-One Fulfillment.

Appendix A describes the Fulfillment Server parameters that can be set in the Fulfillment Server startup script.

Appendix B describes the commands used to operate a Fulfillment Server when it is running.

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 Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Other Information Sources

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of Oracle One-to-One Fulfillment.

If this guide refers you to other Oracle Applications documentation, use only the Release 11*i* versions of those guides.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF). Online help patches are available on MetaLink.

Related Documentation

Oracle One-to-One Fulfillment shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other product documentation when you set up and use Oracle One-to-One Fulfillment.

You can read the documents online by choosing Library from the expandable menu on your HTML help window, by reading from the Oracle Applications Document Library CD included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle Store at http://oraclestore.oracle.com.

Documents Related to All Products

Oracle Applications User’s Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with this release. This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Documents Related to This Product

Implementing Oracle HRMS

This guide describes how to implement Oracle Human Resource Management System.

Using Oracle HRMS — The Fundamentals

Oracle Human Resource Management System is used to create the employees (agents) that will use the business application to submit a fulfillment request. Fulfillment agent groups are created in the Oracle One-to-One Fulfillment administrative console and assigned to a Fulfillment Server.

Oracle Common Application Components Implementation Guide

This guide describes how to implement Oracle Common Application Components, including Resource Manager. The Resource Manager module in Oracle Common Application Components is used to import employees from Oracle Human Resource Management System as resources.

Oracle Common Application Components User Guide

This guide describes how to use Oracle Common Application Components, including Resource Manager. Certain business applications require that employees also be resources. Fulfillment agent groups are created in the Oracle One-to-One Fulfillment administrative console and assigned to a Fulfillment Server.

Oracle Customer Interaction History Implementation Guide

This guide describes how to implement Oracle Customer Interaction History. Oracle Customer Interaction History enables the creation of history records for fulfillment requests. You can use the Oracle One-to-One Fulfillment administrative console to set the interaction history outcomes and results for fulfillment requests that have been processed by the Fulfillment Server.

Oracle Marketing Encyclopedia System Implementation Guide

This guide describes how to implement Oracle Marketing Encyclopedia System. Oracle Marketing Encyclopedia System is a file repository. Oracle Marketing Encyclopedia System stores master documents and queries that are used to specify fulfillment content.

Oracle Marketing Encyclopedia System Concepts and Procedures

This guide describes how to use Oracle Content Manager Marketing Encyclopedia System. The Oracle One-to-One Fulfillment administrative console provides an interface for creating master documents and queries. The master documents and queries stored in Oracle Marketing Encyclopedia System are used by Oracle Collections, Oracle Lease Management, Oracle TeleSales, Oracle TeleService, Oracle Sales Online, Oracle Scripting, and Oracle Student System.

Oracle Content Manager Implementation Guide

This guide describes how to implement Oracle Content Manager. Oracle Content Manager is a file repository. Oracle Content Manager stores templates (similar to master documents in Oracle Marketing Encyclopedia System) and queries that are used to specify fulfillment content.

Oracle Content Manager User Guide

This guide describes how to use Oracle Content Manager. The Oracle One-to-One Fulfillment administrative console provides an interface for creating templates (similar to master documents in Oracle Marketing Encyclopedia System) and queries. The templates and queries stored in Oracle Content Manager are used by Oracle Marketing.

Oracle Marketing Implementation Guide

This guide describes how to implement Oracle Marketing. Oracle Marketing is a campaign management application.

Oracle Marketing User Guide

This guide describes how to upload electronic collateral items (also called deliverables). Collateral may be included in fulfillment request that are created using Oracle Marketing, Oracle Scripting, Oracle Sales Online, and Oracle TeleSales.

Installation and System Administration

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11*i*. It provides a useful first book to read before an installation of Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Oracle Business Intelligence System (BIS), languages and character sets, and Self-Service Web Applications.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications and the technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11*i*. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11*i*. You cannot upgrade to Release 11*i* directly from releases prior to 10.7.

Maintaining Oracle Applications

Use this guide to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the Oracle applications file system and database.

Oracle Applications System Administrator's Guide

This guide provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage concurrent processing.

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It also provides information to help you build your custom Oracle9*i* Forms Developer forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built by using Oracle Forms.

Other Implementation Documentation

Multiple Reporting Currencies in Oracle Applications

This manual details additional steps and setup considerations for implementing the Multiple Reporting Currencies feature.

Multiple Organizations in Oracle Applications

This guide describes how to set up and use the Multiple Organization support feature, so you can define and support different organization structures when running a single installation of Oracle Applications.

Oracle Workflow Guide

This guide explains how to define new workflow business processes as well as customize existing Oracle Applications-embedded workflow processes.You also use this guide to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup and reference information. This manual also provides information on creating custom reports on flexfields data.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Oracle*Metalink*.

Oracle Manufacturing APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes APIs and open interfaces found in Oracle Manufacturing.

Oracle Order Management Suite APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes APIs and open interfaces found in Oracle Order Management Suite.

Oracle Common Application Components Implementation Guide

Many CRM products use components from CRM Application Foundation. Use this guide to correctly implement CRM Application Foundation.

Training and Support

Training

Oracle offers training courses to help you and your staff master Oracle One-to-One Fulfillment and reach full productivity quickly. You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization's structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle One-to-One Fulfillment working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle8*i* server, and your hardware and software environment.

Oracle*MetaLink*

Oracl*eMetaLink* is your self-service support connection with Web, telephone menu, and e-mail alternatives. Oracle supplies these technologies for your convenience, available 24 hours a day, 7 days a week. With Oracl*eMetaLin*k, you can obtain information and advice from technical libraries and forums, download patches, download the latest documentation, look at bug details, and create or update TARs. To use MetaLink, register at (http://metalink.oracle.com).

Alerts: You should check Oracl*eMetaLink* alerts before you begin to install or upgrade any of your Oracle Applications. Navigate to the Alerts page as follows:

Technical Libraries/ERP Applications/Applications Installation and Upgrade/Alerts.

Self-Service Toolkit: You may also find information by navigating to the Self-Service Toolkit page as follows: Technical Libraries/ERP Applications/Applications Installation and Upgrade.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources and customer relationship management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world’s leading supplier of software for information management, and the world’s second largest software company. Oracle offers its database, tools, and applications products, along with related consulting, education, and support services, in over 145 countries around the world.

1 Introduction

Topics include:

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1 Oracle Interaction Center Overview

Every customer interaction — a telephone call, an e-mail message, or a Web chat conversation — presents an opportunity to win new business or improve customer satisfaction. Oracle Interaction Center supports the management and processing of customer relationship activity across all channels of customer contact.

Oracle Interaction Center integrates with several customer relationship business applications in the Oracle eBusiness Suite. This allows access to centralized customer information and business application functionality.

Oracle Interaction Center consists of several products. The modules relating to inbound telephony and outbound telephony are bundled separately. Oracle Interaction Center products include:

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.1 Oracle Advanced Inbound Telephony

Oracle Advanced Inbound Telephony is designed to consistently and effectively handle customer interactions by intelligently routing, queuing and distributing media items. Oracle Advanced Inbound Telephony offers CTI support for market-leading traditional ACD/PBX and IP Telephony platforms, and provides enhanced screen pops on customer data into the Oracle E-Business Suite application. Oracle Advanced Inbound Telephony is fully integrated with Oracle TeleSales, Oracle TeleService and Oracle Collections, thereby minimizing integration time and deployment costs. Oracle Advanced Inbound Telephony also provides the Oracle Telephony Adapter SDK, which can be used to integrate other PBX/ACD and CTI middleware combinations that are not supported by an Oracle telephony adapter.

Oracle Advanced Inbound Telephony is required to telephony enable business applications in the Oracle E-Business Suite. "Telephony-enabled" means that the application can communicate with a telephone system for inbound calls, outbound calls, or both by way of the CTI middleware that handles the messaging between the customer's PBX/ACD and the business application.

The Oracle Advanced Inbound Telephony bundle consists of the following products: Oracle Interaction Center Server Manager, Oracle Universal Work Queue, Oracle Telephony Manager, Oracle Interaction Center Intelligence and Oracle Interaction Blending. Oracle Interaction Center Server Manager is a set of shell scripts and batch files that start, stop and monitor the server processes in an interaction center server group.

See Also

- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.2 Oracle Advanced Outbound Telephony

Oracle Advanced Outbound Telephony is another key part of the Oracle E-Business Suite of applications. It is the module of Oracle Interaction Center that addresses outbound telephony. Oracle Advanced Outbound Telephony consists of two main components:

- A tactical list manager, which determines who to call and when to call them
- An outbound dialing engine, which dials numbers and transfers live contacts to call center agents

Oracle Advanced Outbound Telephony integrates with and relies on Oracle Marketing to create campaigns and lists to execute. Oracle Advanced Outbound Telephony serves as the execution arm for these marketing lists to maximize both outbound list penetration and agent productivity. Oracle Advanced Outbound Telephony also integrates with desktop applications like Oracle TeleSales and Oracle Collections to handle the actual customer interactions. Oracle Advanced Outbound Telephony can be used any time agents need to contact parties via the telephone.

Oracle Advanced Outbound Telephony also integrates with Oracle Customer Interaction History to provide feedback that marketers can use to analyze and measure the success of the marketing campaign, thereby providing a closed-loop marketing process.

Oracle Advanced Outbound Telephony does not include any other telephony management modules, and thus requires the use of Oracle Advanced Inbound Telephony.

See Also

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.3 Oracle Customer Interaction History

Oracle Customer Interaction History provides applications with a common framework for capturing and accessing all "interaction" data associated with customer contacts. Oracle Customer Interaction History acts as a central repository and provides a consistent framework for tracking all automated or agent-based customer interactions.

Applications record interactions through the Oracle Customer Interaction History framework itself, or through other applications that use Oracle Customer Interaction History. This information can be accessed by using the Oracle Customer Interaction History user interface, by using the user interface of an application that is integrated with Oracle Customer Interaction History, or by calling the Oracle Customer Interaction History APIs.

See Also

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.4 Oracle Email Center

Oracle Email Center is a comprehensive solution for managing high volumes of inbound and outbound e-mails. Oracle Email Center reduces the cost per email interaction by automatically replying to certain email inquiries as well as routing others to a skilled set of agents and providing them with a full featured console with cross application functionality.

Oracle Email Center increases customer satisfaction and reduces customer attrition by providing quick, accurate and consistent responses. It also increases agent's efficiency through the use of a full featured, Email Center agent console thereby reducing agent turnover.

An easy-to-use graphical user interface called the Self Service Administration console enables the administrator to configure the system, define processing rules and publish business data, while the Supervisor console enables the supervisor to manage email queues and balance workload. In addition, Oracle Interaction Center Intelligence provides a comprehensive set of reports that enable directors and managers to track email activity and relate it to business events. Through its integration with other Oracle E-Business applications, Email Center provides its agents with cross-application functionality and at the same time provides a utility called Message Component to the business applications for viewing, composing and responding to email messages.

See Also

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.5 Oracle Interaction Blending

Oracle Interaction Blending is a server in an interaction center server group that delivers media (inbound telephony, web callbacks, and outbound telephony) to agents through the Oracle Universal Work Queue work selector according to service levels set in a service plan.

Service levels define the goals for servicing contact channels. Service plans specify the desired level of service for a particular time of day. Service levels are assigned to time intervals in a service plan.

See Also

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.6 Oracle Interaction Center Intelligence

Oracle Interaction Center Intelligence is a Web-based reporting solution that provides intelligent reports that facilitate day-to-day operational and long-term strategic decisions.

The data is presented to the user in a easy-to-use portal format. This format gives the user a unified, role-based, and easily customized view of Interaction Center information. Data presented includes session information, agent productivity metrics and key performance indicators (e.g., speed to answer and abandon rate).

Oracle Interaction Center Intelligence is built on an Oracle proprietary Java-based technology stack (Oracle CRM Foundation, also known as JTT) and a three-tier architecture:

- The first tier consists of the front end (client) which presents the application through an Oracle Applications-compliant Web browser.
- The middle tier is comprised of the Apache Web server and application server, which included as part of the installation of Oracle Applications release 11*i*.
- The third tier represents the database, which is comprised of an Oracle 8*i* or 9*i* database.

See Also

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.7 Oracle One-to-One Fulfillment

Oracle One-to-One Fulfillment provides Oracle E-Business Suite applications with a centralized mechanism for managing fulfillment. Fulfillment is the process of compiling and distributing information to customers.

Oracle One-to-One Fulfillment consists of an API, a server, an administration user interface called the Administration Console, and an agent interface used by Oracle Sales Online. The Oracle One-to-One Fulfillment API is used by the E-Business Suite applications to initiate a request for fulfillment processing. The fulfillment request identifies the agent or server, the content, and the channel. The Oracle One-to-One Fulfillment server processes the request. It compiles the personalized content and determines the recipients of e-mail, fax, print and physical collateral.

Each business application has a unique interface for making a fulfillment request on behalf of one or more parties. Examples of outbound correspondence in E-Business Suite include:

- Notification letters
- Dunning letters
- Lease terms
- Product information
- Survey e-mail invitations

Sales and marketing collateral

See Also

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.8, "Oracle Scripting"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.8 Oracle Scripting

Oracle Scripting provides enterprises with scripts which guide customers, agents and employees through decision flows based on a series of questions and answers. Scripts can be used for a variety of purposes: as traditional interaction center scripts that guide agents through collection of information and provide proactive alerts; as Web-based surveys for customers, prospects, and employees; and as self-service Web scripts that integrate with enterprise Web pages and provide a mechanism to guide Web customers through decision processes. Oracle Scripting is composed of several components: Script Author, the Scripting Engine, the Scripting Administration console, and the Survey Administration console.

Script Author is the development tool with which customized business requirements are translated into miniature programs known as "scripts." Each implementation of Oracle Scripting employs at least one customized script built by Oracle Consulting, consulting partners, or the enterprise. There are various ways in which scripts can be employed to gather or distribute data for an enterprise. For example, a script can serve to unify an agent’s desktop by integrating aspects of various applications, or as a survey questionnaire to solicit specific information from the sample or target population. Script Author offers two ways to create a script; a graphical layout tool and a Script Wizard component.

The Scripting Engine is responsible for displaying the script to the end user, interpreting the end user’s responses to questions and answers, and processing custom code developed in support of the script. The Scripting Engine includes two interfaces (one for agents, and one for executing a script using a Web browser). Any script executed in the Web interface requires survey campaign administration.

The Scripting Administration console provides the user interface with which script developers can launch Script Author as a Java applet, and script administrators can administer Oracle Scripting files, as well as generate, view and analyze a panel footprint report.

The Survey Administration console provides the user interface with which survey administrators establish and maintain survey campaign information, define and manage survey deployments, and view responses from data received.

See Also

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"
- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.9, "Oracle Universal Work Queue"

1.1.9 Oracle Universal Work Queue

Oracle Universal Work Queue provides one click access to work. It brings media queues, prioritized application work queues, work items and messages together in a single access point. Applications integrate with Oracle Universal Work Queue to enable customer channels and to centralize access to application work items. Customer channels deliver media (inbound telephone calls, outbound telephone calls, web callbacks, web collaboration requests and e-mail) to applications through Oracle Universal Work Queue. Applications generate work items (e.g., service requests, leads, opportunities, tasks) and distribute them through assignment, queues and work pools. Oracle Universal Work Queue provides the unifying link between media queues, prioritized application work queues, application work items and Oracle E-Business applications.

See Also

- Section 1.1.1, "Oracle Advanced Inbound Telephony"
- Section 1.1.2, "Oracle Advanced Outbound Telephony"

- Section 1.1.3, "Oracle Customer Interaction History"
- Section 1.1.4, "Oracle Email Center"
- Section 1.1.5, "Oracle Interaction Blending"
- Section 1.1.6, "Oracle Interaction Center Intelligence"
- Section 1.1.7, "Oracle One-to-One Fulfillment"
- Section 1.1.8, "Oracle Scripting"

2

Detailed Product Description

This chapter provides a detailed description of Oracle One-to-One Fulfillment.

Topics include:

- Section 2.1, "Oracle One-to-One Fulfillment Overview"
- Section 2.2, "New in this Release"
- Section 2.3, "Modified in this Release"
- Section 2.4, "Obsolete in this Release"
- Section 2.5, "Concepts"

2.1 Oracle One-to-One Fulfillment Overview

Oracle One-to-One Fulfillment provides Oracle E-Business Suite applications with a centralized mechanism for managing fulfillment. Fulfillment is the process of compiling and distributing information to customers.

Oracle One-to-One Fulfillment consists of an API, a server, an administration user interface called the Administration Console, and an agent interface used by Oracle Sales Online. The Oracle One-to-One Fulfillment API is used by the E-Business Suite applications to initiate a request for fulfillment processing. The fulfillment request identifies the agent or server, the content, and the channel. The Oracle One-to-One Fulfillment server processes the request. It compiles the personalized content and determines the recipients of e-mail, fax, print and physical collateral.

Each business application has a unique interface for making a fulfillment request on behalf of one or more parties. Examples of outbound correspondence in E-Business Suite include:

- Notification letters

- Dunning letters
- Lease terms
- Product information
- Survey e-mail invitations
- Sales and marketing collateral

2.1.1 Basic Terminology

The following terms explain some basic concepts of Oracle One-to-One Fulfillment.

- Cover letter or master document: A document, sent by Oracle One-to-One Fulfillment, which typically includes merge fields that are populated by an associated SQL query.
- Template**:** A high-level definition of one or more Oracle One-to-One Fulfillment master documents, associated queries, and any related collateral.
- Merge field: Data elements in an Oracle One-to-One Fulfillment cover letter or master document that are surrounded by merge field delimiters. In print, merge field delimiters are represented by the double open and close bracket characters < and > (guillemet). These delimiters are represented in HTML by the ASCII codes `&laquo`; (for open or left delimiter) and `&raquo;` (for right or close delimiter). When a fulfillment request runs, any associated query retrieves the merge data from the table columns or views that are specified in the query. For each merge field that is used in a cover letter or master document, there must be one corresponding column name or alias in the select statement of the query to retrieve the appropriate data.
- For Oracle Survey: Invitation or reminder master documents contain at least one merge field to retrieve the survey URL that the recipient must access to participate in the survey. Other merge fields that are often part of invitation and reminder master documents are contact information (title, first name and last name) for each list member, and deployment end date.
- Query: A database query for information that is made in structured query language (SQL). Each Oracle One-to-One Fulfillment cover letter or master document may have an associated query. The information that the query gets is put into placeholders called merge fields in the resulting merged document. A query must have a column name or alias for each data element included as a merge field in the master document. The query may also contain valid column names or aliases for additional data elements, whether or not the data elements are used in each cover letter or master document. For example, you can

construct a single query that requests all information used to populate a survey invitation master document and a survey reminder master document, even if both documents have different merge fields. A single query can be associated with any number of cover letters or master documents. The data from a successfully run query merges with the document.

2.1.2 Templates, Master Documents and Collateral

A template is a grouping of Master Documents and Collateral. (This concept of templates exists only in Oracle One-to-One Fulfillment, not in Marketing Encyclopedia System.) Master Documents are dynamic or static documents. Dynamic content in master documents is defined by merge fields enclosed within special characters. Merge fields are tokens that are replaced by data when fulfillment requests are processed. E-mail and fax master documents can be HTML files or Zip files containing an HTML file plus an image file. Print master documents can be RTF or PDF files.

For RTF documents, merge fields can be entered using the Microsoft Form Field feature. In the form Field Help Text box, the merge field name should correspond to the name or alias used in the SQL query (in uppercase), and should be enclosed within the following text “<?” and “?>”, for example <?CURRENT_DATE?>.

For PDF documents, merge fields can be entered using the Form Tool feature. In the Field Properties box, the merge field Name should correspond to the name or alias used in the SQL query.

Oracle One-to-One Fulfillment Master Documents can optionally have a query associated with them. Some E-Business Suite applications, for example, TeleService or Sales Online, require an associated query.

Collateral (known as Deliverable in Oracle Marketing Online) is a static fulfillment document that does not contain merged data. Collateral are loaded by the Marketing Online Deliverables tab, and may be of any file format, for example, doc, xls, pdf, and so on.

2.1.3 Cover Letters

The cover letter (also known as content item) authoring screen allows the user to create and edit a cover letter based on the content type setup. A cover letter is the primary document that is sent to a customer by e-mail, fax, or print and mail. The user is able to specify a cover letter name, description, and the folder where the cover letter is stored. The following attributes are also available for cover letters:

- Body/Details - The user can enter an HTML body for the cover letter. If the content type has a query associated with it, the user is given a choice to embed the tags or merge fields into the HTML body. The user also has the option to specify a text part of the email. The text part will be read by non-html compliant email clients. Fax requests will only use the HTML part of the content.
- Attachments – The user can search for file attachments from the repository, or upload an attachment from the desktop. By default the attachment is saved to the same location as the cover letter. The user has the option to specify another folder location in which user has the permission to create content.
- Renditions – The One-to-One Fulfillment Oracle print solution supports printing of PDF or RTF documents. These PDF or RTF documents may contain merge fields. The user can select PDF or RTF file from the desktop and upload it as the cover letter content.

2.1.4 E-Business Suite Use of Templates, Master Documents and Collateral

Master Documents only:

- TeleService (seeded master document & query used)
- Lease Management

Collateral only:

- Scripting building block that calls One-to-One Fulfillment

Master Documents and Collateral

- TeleSales

Templates, Master Documents and Collateral:

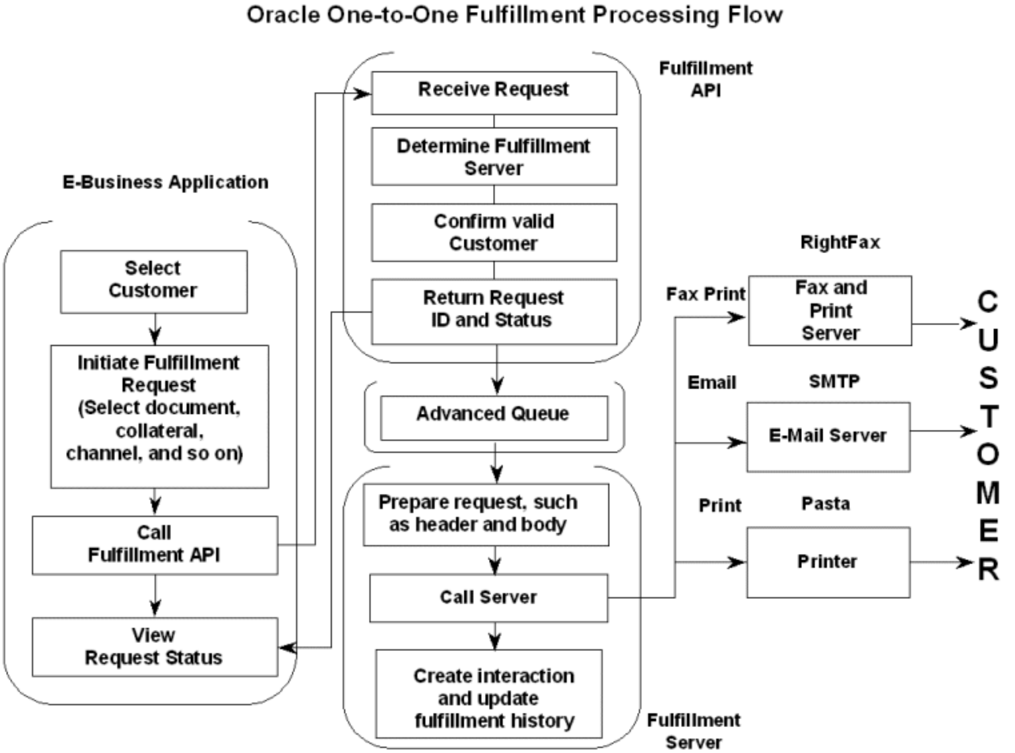
- Sales Online
- Survey
- Collections

Cover Letters

- Oracle Marketing Online
- Student Systems

2.1.5 Process Flow

The following diagram illustrates the process flow of fulfillment requests made with Oracle One-to-One Fulfillment.


Oracle One-to-One Fulfillment Processing Flow
E-Business Application
Select Customer
Initiate Fulfillment Request (Select document, collateral, channel, and so on)
Call Fulfillment API
View Request Status
Fulfillment API
Receive Request
Determine Fulfillment Server
Confirm valid Customer
Return Request ID and Status
Advanced Queue
Prepare request, such as header and body
Call Server
Create interaction and update fulfillment history
Fulfillment Server
RightFax
Fax Print
Fax and Print Server
Email
SMTP
E-Mail Server
Print
Pasta
Printer
CUSTOMER

- The calling applications provide screens or concurrent processes that enable the selection of one or more customers and documents or collateral.
- The Calling Application calls the One-to-One Fulfillment APIs, passing in required data about the particular fulfillment request.
- The Oracle One-to-One Fulfillment APIs process the request and return a request ID (or any errors) to the application. If successful, the API formats and places the request on an Advanced Queue to be picked up by the Fulfillment Server.

- The Fulfillment Server monitors the Request Queue for new requests, and pulls the request from the queue.
- The Fulfillment Server downloads document and collateral from MES or OCM, and sends the request to Java Mail APIs or Fax and Print enablers, or XML Publisher for merging and then prints from Oracle Pasta.
- Oracle One-to-One Fulfillment writes information on the request to Fulfillment History. If flagged to do so by the API, an entry is created in Interaction History is written against the party_id passed in the request.

2.2 New in this Release

The following features and functions are new in this release.

2.2.1 High Quality Oracle Printing

Users can set up Rich Text Format or Adobe Acrobat PDF files, then upload and print them using high quality Oracle Pasta printing.

2.2.2 Malformed E-Mail Address Correction

The capability to view, correct and reprocess malformed e-mail address jobs has been added. The Fulfillment server uses the Java Mail API to partially validate e-mail addresses. An e-mail address containing a blank or null value is listed as malformed. The resulting e-mail messages cannot be sent to the customer.

2.2.3 Job Resubmitting

Documents previously sent to a specific customer can be resubmitted by the same or different output methods.

2.2.4 Bounced E-Mail History Viewing

Bounced e-mail are e-mail messages that have been returned to the sending server, for example, because of an incorrect address, the receiving server is down, or because the recipient's inbox is full. Oracle One-to-One Fulfillment has a feature that shows the total bounced e-mail messages received for each request.

2.2.5 Failed Job Viewing

View details of jobs within requests that failed and the reason for the failure.

2.2.6 Request Pausing and Resuming

When many very large volume e-mail campaigns are running, a backlog may develop because an e-mail server is unable to handle the load, or because an e-mail server goes down. In this situation, an administrator has the option of pausing the request that is being processed, and at a later time resume the request processing from the point where it was paused.

Marketing or Survey e-mail campaigns generally contain content that includes links to further information, promotions, surveys, registration forms, and so on, that are available on external Web servers. If one of these servers is unavailable, then the links embedded within an e-mail message become inaccessible. The administrator may want to pause running the e-mail campaign until the server is available again. When the problem is resolved, the administrator can continue processing the paused request or resubmit the whole request.

Oracle One-to-One Fulfillment offers the ability to pause a “Mass” fulfillment request that has been partially executed (status = in progress), and at a later time allow the administrator to continue the processing using the following commands:

- Resume, which starts the processing from where the request was paused
- Resubmit, which enables the whole request to be reprocessed from the beginning

Details of the jobs within a request that have already been sent or processed are available in the Request History pages.

2.2.7 Oracle Content Manager

Oracle One-to-One Fulfillment uses the Oracle Content Manager Content Type, Cover Letter and Query administration pages, which are accessible through the One-to-One Fulfillment Administration Console menus and offer the following features:

- Cover letter headers, footers and content blocks
- Cover letter versioning
- Utilize folders for storing cover letters
- Uploading content renditions for printing
- Cover letter approvals
- Updating and deleting cover letters

2.2.8 Repeating and Non-Repeating Table Rows With Merge Fields

To accommodate users who want to use HTML tables for document layout, and also to support repeatable fields, from Release 11.5.10 onward Oracle One-to-One Fulfillment can support both repeatable merge field table rows and non-repeatable merge field table rows.

2.3 Modified in this Release

The following features and functions have been modified in this release.

2.3.1 Responsibility

The Responsibility for the Oracle One-to-One Fulfillment Administration Console has been changed from "FMADMIN" to "One-to-One Fulfillment Administrator". Anyone using the FMADMIN Responsibility needs to change it manually to the new Responsibility. The FMADMIN Responsibility has a customized flag that prevents upgrades from updating it automatically.

2.3.2 Search Feature

A search feature is now available in the Status tab > Search sub tab that can search for Requests by the following criteria:

- History Request Identifier
- Customer Identifier
- Customer Name

The results display a summary of the Customer Identifier, Customer Name, History Request Identifier, Job, and Outcome Description. Users can click the History Request Identifier number to view the Request Details page, which displays a summary of the request (server name, submitted & completed times, agent who initiated the request, the outcome code, content sent, statistics, and jobs).

Users can click on the job identifier to view the Job Details page.

2.4 Obsolete in this Release

The following features and functions are obsolete in this release.

2.4.1 Fulfillment Server Parameter Content History

The Fulfillment Server parameter Content History has been deprecated. From Release 11.5.10 onwards, the history of each request is *always* recorded in the Content History table regardless of the setting of this flag. The Fulfillment Server now ignores the server flag -Dhistory.NoContents.

2.4.2 OCM Sub Tabs

The OCM Query sub tab and the OCM Template sub tab, which were under the Template tab, are now obsolete. Their functions have been replaced by the Content tab.

2.5 Concepts

Topics include:

- Integration With Oracle Interaction Center Server Manager
- Advanced Queuing
- E-mail Bounceback
- Fulfillment Requests
- Fulfillment Server
- Fulfillment Channels
- Agent Groups
- Agent Resources
- E-Mail Tracking
- Templates
- MES Master Documents
- Queries
- Data Source
- Collateral

- Fulfillment Request History
- Fulfillment Request Status
- Physical Collateral
- Unsubscribe Option
- Resubmit
- Pause and Resume Request Processing
- Deletion of Sent Fulfillment Requests
- Master Document Editor
- High Quality Printing

2.5.1 Integration With Oracle Interaction Center Server Manager

Oracle Interaction Center Server Manager is part of the Oracle Advanced Inbound HTML administration module. The Oracle Interaction Center Server Manager graphical interface is an efficient way to start, stop and manage all interaction center servers, including the Oracle One-to-One Fulfillment server. This section explains what Oracle Interaction Center Server Manager is and how to integrate it with Oracle One-to-One Fulfillment.

Oracle Interaction Center Server Manager is a Java server process that is installed on each target machine that runs interaction center servers. You can use Oracle Interaction Center Server Manager to do the following Oracle One-to-One Fulfillment processes:

- Set up the Oracle One-to-One Fulfillment Server
- Start and stop the Fulfillment Server
- See the status of the Fulfillment Server
- Inspect the Fulfillment Server Event and Error logs

2.5.2 Advanced Queuing

Advanced Queuing is the message queuing functionality of the Oracle database. Advanced Queuing enables asynchronous communication between applications that use queues. A queue is a repository for messages. Queues are stored in queue tables. Each queue table is a database table and contains one or more queues.

Advanced Queuing supports native XML messages and also allows Advanced Queuing operations to be defined in the XML-based Internet-Data-Access-Presentation (IDAP) format. IDAP, an extensible message invocation protocol, is built on Internet standards, using HTTP and e-mail protocols as the transport mechanism, and XML as the language for data presentation.

Business applications submit the fulfillment data in accordance with the Oracle One-to-One Fulfillment API procedures, which format the fulfillment request as an XML message. The API places fulfillment requests in a queue (enqueuing) and the Fulfillment Server retrieves fulfillment requests from a queue (dequeuing).

For more information about Advanced Queuing, see *Oracle 9i Application Developer's Guide - Advanced Queuing*.

2.5.3 E-mail Bounceback

Oracle One-to-One Fulfillment supports e-mail bounceback processing, the process of returning e-mail to the sending server because of an incorrect address (hard bounceback) or because the recipient's inbox is full (soft bounce back).

To enable bounceback processing, see

2.5.4 Fulfillment Requests

A fulfillment request is the data object that is submitted for fulfillment processing. The E-Business Suite application uses the Oracle One-to-One Fulfillment application programming interface (API) to specify the components of the content (for example, a master document from Oracle Marketing Encyclopedia System), the output channel (e-mail, fax, or print), and the user of Fulfillment Server. The resulting data object is placed in the fulfillment request queue. The fulfillment request queue is provided by Advanced Queuing.

Each E-Business Suite application has a unique interface for making a fulfillment request on behalf of one or more parties. For information about how to initiate a fulfillment request, see the user guide or online help for the business application.

Content

The following table briefly describes the different type of content. For a more detailed description, see the relevant section.

Content Type	Location	Description	Set Up in Oracle One-to-One Fulfillment?
Master Document	Oracle Marketing Encyclopedia System	A static or dynamic HTML, Zip, RTF or PDF document.	Yes
Query	Oracle One-to-One Fulfillment	A SQL statement that provides data for a dynamic master document	Yes
Collateral	Oracle Marketing	A static document of any type	No
Template	Oracle One-to-One Fulfillment	A grouping of master documents and collateral documents.	Yes
Cover Letter	Oracle Content Manager	A static or dynamic HTML, Zip, RTF or PDF document.	Yes
Query	Oracle Content Manager	A SQL statement that provides data for a dynamic template.	Yes

2.5.5 Fulfillment Server

Fulfillment Server is a multi-threaded Java server that processes fulfillment requests by de-queuing the request from an Advanced Queue. Business applications use Fulfillment APIs to place fulfillment requests in the queue.

You can specify the time during which the server polls the fulfillment request queues in the database. It does not physically start or stop the server. If the server is running, the server will continue to process retrieved fulfillment requests even when it is not polling the database.

When a fulfillment request is processed, it creates a report in the Fulfillment history and an interaction record in Oracle Customer Interaction History. You can select which outcome reason and result you want to associate with a successful or failed fulfillment request.

2.5.6 Remote Commands

The remote command script lets users interact with the Fulfillment Server without the need to maintain the original command prompt that was used to start the server. To issue a command, type the name of the remote command script and the command.

The Fulfillment Server is configured in the startup script to monitor a port for remote commands. The port is specified in both the startup script for the server and the remote command script. The remote command script runs on the same computer as the Fulfillment Server or on a computer that is on the same network as the Fulfillment Server.

When a command is sent through the remote command script, the computer that issued the command terminates activity after it receives a response to the command. To prevent unnecessary workload for the Fulfillment Server, the computer that issues the remote command retains a connection to the Fulfillment Server only for the duration of the remote command.

2.5.7 Fulfillment Channels

The Fulfillment Server compiles the content and sends the content to an output device. Output include e-mail, fax and print.

For e-mail fulfillment requests, Oracle One-to-One Fulfillment supports any e-mail server that uses SMTP (Simple Mail Transfer Protocol) to transmit e-mail messages. For fax fulfillment requests, Oracle One-to-One Fulfillment supports Captaris RightFax and Oracle Pasta printing.

One or more e-mail servers can be associated with a Fulfillment Server.

2.5.8 Agent Groups

An agent is an employee who uses Oracle software. For agent using an E-Business Suite application to initiate a Fulfillment request, create groups of Fulfillment agents in the Oracle One-to-One Fulfillment Administration Console, and then assign them to a group.

2.5.9 Agent Resources

A resource is an employee whose record has been imported into the Oracle CRM Resource Manager, which provides a resource ID that makes that person a resource who is authorized to use CRM applications that require a resource ID for Customer Interaction History recording.

2.5.10 E-Mail Tracking

Oracle One-to-One Fulfillment tracks the outcomes of e-mail sent by a batch request, single request, or mass request. The Status page for the batch request shows the following statistics:

- Sent
    - Opened

  - Unopened
- Not Sent
  - Malformed
  - Do not contact
  - Other failures
- Bounced
- Unsubscribed
- Resubmitted Malformed

The Fulfillment Server dispatches the e-mail requests to all the recipients of the e-mail. If Track Email is enabled, when a recipient opens the e-mail to read, at the end of the e-mail is an invisible reference to an image on a Web server. As the e-mail is downloaded, when a reference to the hidden image is encountered, the client tries to download the image from that particular Web server. However, the hidden image tag references a Java Server Page (JSP), which is responsible for updating the required tables with information that specifies which recipient had read the e-mail. The JSP that handles the hidden image needs to be accessible outside the firewall.

The information that is required to decide which recipient has opened the e-mail is present as a part of the query string in the URL. The JSP parses all the parameters from the query string and updates the database based on the request query string of the request URL. Also, the JSP is responsible for returning a substitute image instead of the one that is requested originally. The substitute image is a transparent image.

2.5.11 Templates

A template is a grouping of material that is useful but not required for distributing master documents and collateral. Collateral is static content.

An template is a package of content that is compiled by the Fulfillment Server. The fulfillment request identifies the template being used to compile the content.

2.5.12 MES Master Documents

A master document is a message sent from Oracle One-to-One Fulfillment. It is an HTML file or Zip file that contains the main content in a fulfillment item.

Examples of master documents include:

- Cover letters
- Service request notifications
- Past due notices
- Lease contracts
- Sales offers

A master document can be static or dynamic. A static master document does not have merge fields, so that each recipient gets the same content. A dynamic master document uses merge fields populated by an associated SQL query to combine information from the database (such as the name of the recipient or account information) into the content, so that each recipient gets personalized content. See Queries for information about queries.

The way in which the E-Business Suite application uses a master document determines whether the master document is used as a standalone or as a part of a template. See Templates for information about templates.

A master document can be created in an HTML editor or word processing application, but it must be saved as an HTML file. It can be uploaded as an HTML file or in a ZIP file.

When editing HTML files in a word processor, remove from the HTML code any propriety information that was inserted by the word processor, otherwise the server will not process the document. For example, saving a Microsoft Word document as HTML inserts Microsoft proprietary information in the `<HTML>` header tag, that is, a line in the header stating,

```
<META NAME ="Generator" CONTENT="Microsoft Word 97>
```

To remove such propriety information, use a generic text editor such as Microsoft Notepad or WordPad.

2.5.13 Oracle Content Manager Profile Options

Oracle Content Manager has two profile options relevant to Oracle One-to-One Fulfillment, data security and the New Content Items folder.

Data Security

The data security profile option enables and disables security in Oracle Content Manager. By default this option is disabled. To enable folder permissions, do the following:

- Set the security profile to Y.
- Set the necessary folder permissions for users of Oracle One-to-One Fulfillment.

New Content Items Folder

Oracle Content Manager stores new content items in a folder that you need to specify. If these profile options are not set, Content Items cannot be submitted properly.

The following table describes the data security profile options.

Table 2–1 Data Security Profiles

Profile Name	Default Value	Comments
IBC: Use Access Control	N	Data security is disabled. Any user can create, update or delete a content item. All content items created are approved immediately. If set to Y, data security is enabled. A check is made to see if the user has appropriate permissions to perform the actions.
IBC: Folder/Directory for new items	COMMON	Specifies the folder where new content items will be created. **Note**: When data security is enabled, the user must have the required permission on this folder to create, update and delete items in this folder. Permissions can be granted from the Oracle Content Manager Foldering tab.

See Also

Oracle Content Manager User Guide

2.5.14 Queries

A query is a SQL statement that extracts selected data from a database. A query determines:

- Which records meet the defined criteria.
- Which fields of those records the query will return.

The returned data is merged with a master document to create a customized e-mail, fax, or printed mail for each recipient.

A single query can be associated with more than one master document if the merge field names used in the document match the database column names or aliases.

Use the following syntax when creating a SQL query.

```
SELECT <column name> <alias>,
        <column name> <alias>,
        <column name> <alias>,
        <column name> <alias>
FROM <table name>
WHERE <column name> = :<bind variable>
```

When creating a query, note the following conditions:

- Aliases are optional.
- SQL statements typically end with a semicolon (;). However, in this case, the query engine appends the semicolon. Do NOT append a semicolon to the end of this query.
- Column name aliases follow the column names and appear before the comma or the next keyword. Use aliases as substitutes in master documents for the actual column names. Aliases are not required, but can be helpful.
- The table from which the columns are selected follows the FROM keyword.
- The bind variable specifies which row is pulled from the table to give the actual data for each merged document. A colon in front of a bind variable makes it a global variable. This enables the query to recognize it after the value is passed from the user interface. For example:

  ```
  SELECT person_last_name last_name,
      party_id customer_number,
      creation_date,
      phone_number
  FROM jtf_contact_points_v
  WHERE person_first_name = :fname;
  ```

Merge fields should match exactly with the Master Document.

2.5.15 Data Source

Creating a data source file is an optional method of saving queried data to be used in a master document. A data source file is a text file that contains tab-separated merge fields that have been obtained by a query. Use the data source file to obtain the columns available from the query so that they can be used as merged fields. This can be helpful when creating master documents. The master document must be an HTML file.

If you enter the previous sample query as the query string, then the data source utility pulls the aliases rather than the column names when it creates the data source file.

The data source utility opens a window that shows the following column headers in a plain background window.

```
lname    CustNumb    creation_date    phone_number
```

2.5.16 Collateral

Use Oracle Marketing to upload collateral items. Collateral may be included in fulfillment requests that are created using Oracle Scripting, Oracle Sales Online, and Oracle TeleSales.

2.5.17 Fulfillment Request History

The Status tab > History sub tab is a record of fulfillment requests that a Fulfillment Server is currently being processed or has been processed, successfully or unsuccessfully. You can view the details of a fulfillment request, re-submit a fulfillment request, and delete a history entry from the history list.

To view requests by outcome, select an outcome level from the View list. The listed outcomes are defined as follows:

- All: All Fulfillment requests are displayed in descending order of Request ID.
- Cancelled: Fulfillment Requests that have been canceled by the administrator or by the calling application.
- Failed: Fulfillment Requests that were not processed successfully. If the request contains more than one job, all jobs within the request failed.
- Partially Successful: Fulfillment Requests that contain one or more jobs that could not be processed successfully, and at least one job that was successfully processed. For example, if a customer is listed as "do not contact" for the particular media channel, then a failure is logged for that job and the outcome of the request becomes "partially successful."
- Stalled: Fulfillment Requests that have stalled while being processed by the Fulfillment Server. A request can stall due to many reasons, but if it is stalled

because of a database connectivity issue, the server will continue to periodically try to reconnect and, if successful, it will continue to process the request. Most stalled requests are caused by the fulfillment server stopping abnormally, in this case restart the server to continue processing the request.

- Successful: Fulfillment Requests that were processed successfully.

2.5.18 Fulfillment Request Status

The Status sub tab provides access to the status of fulfillment requests that have been placed in the request queue, but have not yet been retrieved by the Fulfillment Server. You can view the status of a fulfillment request, cancel a fulfillment request, resubmit the request, and delete a status entry from the status list.

Each status is defined as follows:

- Submitted: Fulfillment Requests that have been submitted to the Fulfillment Server for processing
- Re-submitted: Fulfillment Requests that have been re-submitted for processing
- In Progress: Fulfillment Requests that are currently being processed
- Paused: Fulfillment requests that have been partially processed and then paused
- Resumed: Fulfillment requests that were paused and subsequently resumed

2.5.19 Physical Collateral

APIs enable E-Business Suite applications to pass physical collateral requests. In turn, One-to-One Fulfillment integrates with Order Management to execute these requests.

2.5.20 Unsubscribe Option

One-to-One Fulfillment has an unsubscribe feature for e-mail fulfillment requests. The e-mail contains an embedded URL link to a Web page so that when users accesses the Web page, a validity check is performed and users have the option to unsubscribe.

The Unsubscribe option in the Fulfillment Server configuration causes the Fulfillment Server to insert a hypertext link into the content of e-mail that will allow e-mail recipients to modify their contact preferences.

For e-mail requests, the outgoing e-mail contains an embedded URL to a Web page. The URL leads to the Oracle Marketing Online Contact Preferences page, where the recipient can set contact channels. The format of the unsubscribe link is:

```
http://<host>:<port>/OA_HTML/<name of the
jsp>?PAGE.OBJ.requestId=&PAGE.OBJ.objId=
```

The request_id is the numerical identifier for the fulfillment request. If the recipient elects to unsubscribe from e-mail contact or all contact, then Oracle Marketing Online uses the request identifier to update Oracle One-to-One Fulfillment History with this information.

The party_id is the numerical identifier of the recipient of the e-mail. The party_id is used when the amsContactPreferencesMain.jsp page is called to confirm that the party is valid, and that the appropriate Contact Preferences table within TCA can be modified when the customer unsubscribes. The party identifier is encrypted so that it is not visible to the e-mail recipient. Oracle Marketing Online uses the encryption key in the JTO: Security: Merge Field Encryption Key profile option to decrypt the party identifier and enable access to the party's contact preference settings.

When the recipient accesses the Web page, the recipient can unsubscribe from e-mail. This action is recorded in Oracle Customer Interaction History and, if e-mail tracking is enabled, is reflected in the e-mail statistics for the batch or mass e-mail fulfillment request.

To enable or disable the unsubscribe link, use the Unsubscribe box on the E-mail Server page for the Fulfillment Server. This option affects the e-mail fulfillment requests that are processed by all e-mail servers assigned to the Fulfillment Server. It cannot be set for an individual e-mail server.

2.5.21 Resubmit

The One-to-One Fulfillment Administration Console has a Resubmit option in the Status tab > History sub tab. This options enables a request to be resubmitted by using the original data.

The One-to-One Fulfillment Administration Console provides a Resubmit option for jobs in the Jobs Details page. This options enables a job (that was sent to a single customer) to be resubmitted using the original data. The user is able to select an alternative delivery channel and contact address for the recipient.

2.5.22 Pause and Resume Request Processing

One-to-One Fulfillment has a pause and continue function that administrators can use to pause an in-progress batch request, and later resume processing the request. This function is available in the Request Status page.

2.5.23 Deletion of Sent Fulfillment Requests

A Delete option is in the Status tab > History sub tab for deleting canceled, completed, and failed requests.

2.5.24 Master Document Editor

The One-to-One Fulfillment Master Document Editor simplifies the administrators' job of creating and publishing static and dynamic documents for communicating with customers.

The Master Document editor provides a fast and simple mechanism for creating master documents, creating and associating a query and uploading into the Management Encyclopedia System.

Text can be written in Rich Text Format, which supports multiple fonts, sizes and styles, and URL links and image insertion.

2.5.25 High Quality Printing

The high quality print Oracle solution enables the uploading of RTF or PDF Form documents, and offers E-Business Suite Applications the ability to initiate requests to print RTF or PDF Form documents or cover letters.

An Administrator creates an RTF document using Microsoft Word (or opens an existing document), uses the Form feature to insert the relevant personalized data (merge fields) in the appropriate positions in the letter or table, and then saves the document as an RTF file. When the document has been uploaded into the Content Repository, it is available for sending. Static PDF or RTF documents, such as attachments, can be printed in addition to the primary personalized document.

Oracle One-to-One Fulfillment also supports printing documents and collateral by way of Captaris RightFax.

2.5.25.1 Oracle Pasta

Oracle Pasta is an Oracle printing solution that uses an executable called FNDPSTAX. The FNDPSTAX tool submits print requests to the operating system printing functions:

- lp on UNIX and Linux
- print on Windows

To use the High-Quality Printing feature, you must first be able to print from the Fulfillment Server host machine using either lp or print, depending on your operating system. It is the responsibility of the local system administrator to configure the operating system and network for printing.

2.5.25.2 XPDF pdftops

XPDF is an open source viewer for PDF files. It includes a PDF-to-Post Script conversion tool called pdftops. Pasta uses PDF to PS to convert the resulting PDF files into PS before spooling them to the printer. Download XPDF at http://www.foolabs.com/xpdf/ for a download.

2.5.25.3 XML Publisher

XML Publisher is an Oracle toolset that provides utilities to transform RTF and PDF documents into a format called XSL-FO (or eXtensible Stylesheet Language-Formatting Objects). The XSL-FO format is used as a base format that can be transformed back into RTF or PDF (and in later XML Publisher versions) HTML. XSL-FO is not intended for end users like marketers, survey creators, or other Fulfillment users who are responsible for creating Fulfillment Content. It is mentioned here for information purposes only. XML Publisher 4.3 contains the necessary RTF conversion utilities.

2.5.25.4 RTF and PDF

Due to the reliance on the XML Publisher utilities, only Fulfillment requests that contain strictly RTF or PDF documents can be printed through the High-Quality Print solution. Requests that contain any other file types can only be printed by the existing RightFax printing solution.

3 Before You Begin

This chapter describes the system requirements and product dependencies for Oracle One-to-One Fulfillment. It also describes how to access the administrative console for Oracle One-to-One Fulfillment.

Topics include:

- Section 3.1, "System Requirements"
- Section 3.2, "Mandatory Dependencies"
- Section 3.3, "Conditional Dependencies"
- Section 3.4, "Related Documentation"
- Section 3.5, "Installing Oracle One-to-One Fulfillment"
- Section 3.6, "Accessing Oracle One-to-One Fulfillment"

3.1 System Requirements

This section describes the system requirements for Oracle One-to-One Fulfillment.

Topics include:

- Section 3.1.1, "Fulfillment Server"
- Section 3.1.2, "Output Servers"

3.1.1 Fulfillment Server

The Fulfillment Server processes the fulfillment requests that have been submitted by the business applications. All of the files required to use the Fulfillment Server are installed when Oracle Applications is installed. You can run the Fulfillment Server on the applications server. You can also run the Fulfillment Server from a separate computer that has access, by a network, to the Oracle Applications environment and database. If you choose to run the Fulfillment Server on another computer, then you must provide the following supporting files on the computer:

- Java Files
- Java Development Kit (JDK) or Java Runtime Environment (JRE)
- Database Connectivity File

3.1.1.1 Java Files

You need access to the following Oracle Applications Java files to run the Fulfillment Server:

- apps.zip
- jdbc111.zip or jdbc12.zip

> **Note:** If you are using JDK or JRE 1.1.8, then put jdbc111.zip before jdbc12.zip in the CLASSPATH. If you are using JDK or JRE 1.2 or higher, then put jdbc12.zip before jdbc111.zip in the CLASSPATH. See Section 3.1.1.2, "Java Development Kit (JDK) or Java Runtime Environment (JRE)" for information about JDK and JRE.

- xmlparserv2.zip

The apps.zip file contains the instructions for the Fulfillment Server. The apps.zip file is maintained in AU_TOP, the path to the Application Utilities directory, and copied to JAVA_TOP, the path to the java directory, for runtime operation. The same file exist in both the AU_TOP and JAVA_TOP directories.

The jdbc111.zip and jdbc12.zip files enable Java programs to access the Oracle database server. The xmlparserv2.zip file is the XML Parser library. These files are maintained in JAVA_TOP.

If the Fulfillment Server is run on the Oracle Applications server, then you can use the Java files on the Oracle Applications server. If the Fulfillment Server is run on another computer, then you will have to point to the Java files on the Oracle Applications server or copy the Java files to the computer.

If you point to the Java files, then patches to the Java files are immediately available to the computer. In addition, system processing may be slower. If you copy the Java files to the computer, patches to the Java files on the Oracle Applications servers are not immediately available to the computer. However, system processing may be faster.

You have the following options:

- Create a network file system link to the Java files on the Oracle Applications server to which patches are most often applied, such as the Administration Server. The format of the path to the files will depend on how the network file system is set up. For information specific to your enterprise, please contact your system administrator.
- Perform a Web tier installation of Oracle Applications. This will automatically copy the Java files.
- Manually copy the Java files from an Oracle Applications server to the computer.

The location of the files must be specified in the CLASSPATH. The CLASSPATH is a parameter in the adovars.env file, a file that is called from the main applications environment file (<dbname>.env). It lists paths to the files that are scanned for Java class files needed at runtime.

You can override a path in the CLASSPATH parameter by specifying a different path in the startup script for the Fulfillment Server. In the following example, the line specifies the path to the apps.zip file on a computer other than the Oracle Applications server.

```
set CLASSPATH=%CLASSPATH%;D:\javafiles\apps.zip
```

3.1.1.2 Java Development Kit (JDK) or Java Runtime Environment (JRE)

The Java Development Kit (JDK) or Java Runtime Environment (JRE) provide the Java Virtual Machine in which the Fulfillment Server runs. Java Runtime Environment (JRE) is installed during the installation of the Oracle Applications Web server. JRE is found in JAVA_TOP, the path to the java directory.

If the Fulfillment Server is run on the applications server, then you can use the JRE on the Oracle Applications Web server. If the Fulfillment Server is run on another computer, then you will have to install JDK or JRE on the computer.

To install JDK or JRE on another computer, you have the following options:

- Perform a Web tier installation of Oracle Applications on the target computer. This will automatically install JRE.
- Obtain JDK or JRE and manually install it and on the target computer.

You can download JDK or JRE from Sun or the hardware vendor. You should perform all necessary operating system tuning for the specific platform. For example, certain Hewlett-Packard operating systems require that the operating system kernel parameter max_proc_thread be raised from the default value of 64 to at least 1024. Please contact your hardware vendor for additional information.

You can use the following JDK or JRE versions to run the Fulfillment Server:

- 1.3.1
- 1.3.0
- 1.2.2
- 1.1.8

> **Note:** If you are using JDK or JRE 1.1.8, then put jdbc111.zip before jdbc12.zip in the CLASSPATH. If you are using JDK or JRE 1.2 or higher, then put jdbc12.zip before jdbc111.zip in the CLASSPATH. See Section 3.1.1.1, "Java Files" for information about Java database connectivity files.

The location of the certain zip and jar files in the JDK or JRE must be specified in the CLASSPATH. The CLASSPATH is a parameter in the adovars.env file, a file that is called from the main applications environment file (<dbname>.env). It lists paths to the files that are scanned for Java class files needed at runtime.

You can override a path in the CLASSPATH parameter by specifying a different path in the startup script for the Fulfillment Server. In the following example, the line specifies the path to the classes.zip file on a computer other than the Oracle Applications server.

```
set CLASSPATH=%CLASSPATH%;D:\jdk1.1.8\lib\classes.zip
```

3.1.1.3 Database Connectivity File

The database connectivity file (*.dbc) is a text file that contains the connection parameters for the Oracle Applications database. A database connectivity file is automatically created when Oracle Applications is installed.

There can be more than one database connectivity file. While a database connectivity file can have any name (*.dbc), the typical naming convention is <host_

name>_<dbname>.dbc. These files are usually found in FND_TOP in the secure directory (FND_TOP/secure/<host_name>_<dbname>.dbc).

If the Fulfillment Server is run on the Oracle Applications server, then you can use the database connectivity file that is created when Oracle Applications is installed. If the Fulfillment Server is run on another computer, then you will have to put a database connectivity file on the computer. You can copy the database connectivity file or manually create one. The location of the database connectivity file must be specified in the startup script for the Fulfillment Server.

The following table lists the necessary connection parameters for the database connectivity file. If you do not know the appropriate values, then contact your Oracle Applications system administrator or database administrator.

Parameter	Description
TWO_TASK	The name of the database or the SID.
FNDNAM	The default value is 'apps.' Do not change this value.
GWYUID	The gateway userid and password.
FND_MAX_JDBC_CONNECTIONS	The default value is '20.' Do not change this value.
GUEST_USER_PWD	An Oracle Applications database guest userid and password.
APPS_JDBC_DRIVER_TYPE	The default value is 'THIN.' Do not change this value.
DB_HOST	The name of the host computer for the Oracle Applications database.
DB_PORT	The listener port number for the Oracle Applications database. The Oracle database client on the target computer communicates with the Oracle Applications database using this port number.

The following is an example of a DBC file:

```
#DB Settings
#10APR2003
TWO_TASK=dbsid
FNDNAM=apps
GWYUID=applsyspub/pub
GUEST_USER_PWD=userid/password
APPS_JDBC_DRIVER_TYPE=THIN
DB_HOST=machinename.domainname.com
DB_PORT=1527
```

3.1.2 Output Servers

The Fulfillment Server sends fulfillment content to an output device. Output devices include e-mail, fax, and print servers. In addition to the standard technology that is installed with Oracle Applications, the following technology is required to successfully implement Oracle One-to-One Fulfillment:

- For e-mail output, any outgoing e-mail server that complies with SMTP (Simple Mail Transfer Protocol), for example, UNIX Sendmail
- For fax or print output, Captaris RightFax Enterprise Server

See Appendix 5, "Integration Tasks" for information about integrating Captaris RightFax Enterprise Server with Oracle One-to-One Fulfillment.

3.2 Mandatory Dependencies

Mandatory dependencies are required for Oracle One-to-One Fulfillment to function. Oracle One-to-One Fulfillment relies on several products for data and functionality. The following products must be installed and implemented before you begin the implementation of Oracle One-to-One Fulfillment:

- Oracle Human Resource Management System
- Oracle Common Application Components
- Oracle Customer Interaction History
- Oracle Marketing Encyclopedia System

  Required for use with Oracle Collections, Oracle Lease Management, Oracle TeleSales, Oracle TeleService, Oracle, Sales Online, Oracle Scripting, and Oracle Student System.
- Oracle Content Manager

  Required for use with Oracle Marketing.
- Oracle Marketing

  Required for use with Oracle Marketing, Oracle Scripting, Oracle Sales Online, and Oracle TeleSales.
- Business Applications

Oracle Human Resource Management System

Oracle Human Resource Management System is used to create the employees (agents) that will use the business application to submit a fulfillment request. Fulfillment agent groups are created in the Oracle One-to-One Fulfillment administrative console and assigned to a Fulfillment Server.

For information about Oracle Human Resource Management System, see *Implementing Oracle HRMS* and *Using Oracle HRMS — The Fundamentals*.

Oracle Common Application Components

Certain business applications require that employees also be resources. The Resource Manager module in Oracle Common Application Components is used to import employees from Oracle Human Resource Management System as resources. Fulfillment agent groups are created in the Oracle One-to-One Fulfillment administrative console and assigned to a Fulfillment Server.

For information about Oracle Common Application Components, see *Oracle Common Application Components Implementation Guide* and *Oracle Common Application Components User Guide*.

Oracle Customer Interaction History

Oracle Customer Interaction History enables the creation of history records for fulfillment requests. You can use the Oracle One-to-One Fulfillment administrative console to set the interaction history outcomes and results for fulfillment requests that have been processed by the Fulfillment Server.

For information about Oracle Interaction History, see *Oracle Interaction History Implementation Guide*.

Oracle Marketing Encyclopedia System

Oracle Marketing Encyclopedia System stores master documents and queries that are used to specify fulfillment content. The master documents and queries stored in Oracle Marketing Encyclopedia System are used by the following business applications:

- Oracle Collections
- Oracle Lease Management
- Oracle TeleSales
- Oracle TeleService
- Oracle Sales Online

- Oracle Scripting
- Oracle Student System

The Oracle One-to-One Fulfillment administrative console provides an interface for

- Creating master documents using a document editor and uploading those master documents to Oracle Marketing Encyclopedia System
- Uploading externally created master documents to Oracle Marketing Encyclopedia System
- Creating queries in Oracle Marketing Encyclopedia System

For information about Oracle Marketing Encyclopedia System, see *Oracle Marketing Encyclopedia System Implementation Guide* and *Oracle Marketing Encyclopedia System Concepts and Procedures*.

Oracle Content Manager

Oracle Content Manager stores templates (similar to master documents in Oracle Marketing Encyclopedia System) and queries that are used to specify fulfillment content. The templates and queries stored in Oracle Content Manager are used by Oracle Marketing.

The Oracle One-to-One Fulfillment administrative console provides an interface for

- Creating templates using a document editor and uploading those templates to Oracle Content Manager
- Uploading externally created templates to Oracle Content Manager
- Creating queries in Oracle Content Manager

For information about Oracle Content Manager, see *Oracle Content Manager Implementation Guide* and *Oracle Content Manager User Guide*.

Oracle Marketing

Oracle Marketing is used to upload collateral items. Collateral may be included in fulfillment requests that are created using Oracle Marketing, Oracle Scripting, Oracle Sales Online, and Oracle TeleSales.

For information about Oracle Sales Online, see *Oracle Marketing Implementation Guide* and *Oracle Marketing User Guide*.

Business Applications

Fulfillment requests originate from a business application. The following business applications integrate with Oracle One-to-One Fulfillment:

- Oracle Collections
- Oracle Lease Management
- Oracle Marketing
- Oracle Scripting
- Oracle Sales Online
- Oracle Student System
- Oracle TeleSales
- Oracle TeleService

Oracle Collections Oracle Collections is a debt recovery application. Agents use Oracle Collections to manage delinquent accounts and collect payments. Agents also use Oracle Collections to send correspondence, such as a request for payment or an account statement, to a party.

Oracle Collections uses Oracle One-to-One Fulfillment to process single or batch requests by e-mail, fax, and print output channels. The content of the correspondence is stored in Oracle Marketing Encyclopedia System as reusable master documents. The master documents can be personalized using merge fields and queries.

Oracle Collections generates a fulfillment request when an automatic work item is generated or an agent chooses to send a master document. Oracle Collections also records the status of the fulfillment request in Oracle Customer Interaction History.

For information about Oracle Collections, see *Oracle Collections Implementation Guide* and *Oracle Collections User Guide*.

Oracle Lease Management Oracle Lease Management is a asset management application. Agents use Oracle Lease Management to manage the leases for assets. Agents also use Oracle Lease Management to send correspondence, such as the terms of a lease, to a party.

Oracle Lease Management uses Oracle One-to-One Fulfillment to process single requests by e-mail. The content of the correspondence is stored in Oracle Marketing Encyclopedia System as reusable master documents. The master documents can be personalized using merge fields and queries.

Oracle Lease Management generates a fulfillment request when an agent chooses to send a master document. Oracle Lease Management also records the status of the fulfillment request in Oracle Customer Interaction History.

For information about Oracle Lease Management, see *Oracle Lease Management Implementation Guide* and *Oracle Lease Management User's Guide*.

Oracle Marketing Oracle Marketing is a campaign management application. Marketing professionals use Oracle Marketing to define, execute, and monitor marketing campaigns. Marketing professionals also use Oracle Marketing to send correspondence, such as an offer, to marketing targets.

Oracle Marketing uses Oracle One-to-One Fulfillment to process single or batch requests by e-mail. The content of the correspondence is stored in Oracle Content Manager as reusable template (similar to master documents in Oracle Marketing Encyclopedia System). The templates can be personalized using merge fields and queries. Collateral items (also called deliverables) are uploaded using Oracle Marketing.

Oracle Marketing generates a fulfillment request when a user chooses to send a template. Oracle Marketing also records the status of the fulfillment request in Oracle Customer Interaction History.

For information about Oracle Marketing, see *Oracle Marketing Implementation Guide* and *Oracle Marketing User Guide*.

Oracle Scripting Oracle Scripting is a script and survey management application. Survey administrators use Oracle Scripting to define, execute, and monitor surveys. Survey administrators also use Oracle Scripting to send correspondence, such as invitations and reminders, to survey targets.

Oracle Scripting uses Oracle One-to-One Fulfillment to process batch requests by e-mail. The content of the correspondence is stored in Oracle Marketing Encyclopedia System as reusable master documents. The master documents can be personalized using merge fields and queries. Optional collateral items (also called deliverables) are uploaded using Oracle Marketing.

Oracle Scripting generates a fulfillment request when the survey campaign is deployed. Oracle Scripting also records the status of the fulfillment request in Oracle Customer Interaction History.

For information about Oracle Scripting, see *Oracle Scripting Implementation Guide* and *Oracle Scripting User Guide*.

Oracle Sales Online Oracle Sales Online is a sales management application. Sales representatives use Oracle Sales Online to manage sales leads and opportunities. Agents also use Oracle Sales Online to send correspondence, such as a quote, to a party.

Oracle Sales Online uses Oracle One-to-One Fulfillment, in the form of a wizard, to process single requests by e-mail and fax. The content of the correspondence is stored in Oracle Marketing Encyclopedia System as reusable master documents. The master documents can be personalized using merge fields and queries. Collateral items (also called deliverables) are uploaded using Oracle Marketing. A template is used to join master documents and collateral item in a single package.

Oracle Sales Online generates a fulfillment request when an agent chooses to compile a templates that contains a master document and collateral items. Oracle Sales Online also records the status of the fulfillment request in Oracle Customer Interaction History.

For information about Oracle Sales Online, see *Oracle Sales Online Implementation Guide* and *Oracle Sales Online User Guide*.

Oracle Student System Oracle Student System is a student information management application. School officials use Oracle Student System to manage the entire life cycle of a student from inquiry through graduation. School officials also use Oracle Student System to send correspondence, such as a notifications and statements, to a party.

Oracle Student System uses Oracle One-to-One Fulfillment to process single or batch requests by e-mail, fax, and print output channels. The content of the correspondence is stored in Oracle Marketing Encyclopedia System as reusable master documents. The master documents can be personalized by using merge fields and queries.

Oracle Student System uses concurrent processes to generate a fulfillment requests. Oracle Student System also records the status of the fulfillment request in Oracle Customer Interaction History.

For information about Oracle Collections, see *Oracle Student System Implementation Guide* and *Oracle Student System User's Guide*.

Oracle TeleSales Oracle TeleSales is a contact management application. Agents use Oracle TeleSales to manage inbound telesales and outbound telemarketing. Agents also use Oracle TeleSales to send correspondence, such as cover letters or collateral items, to a party.

Oracle TeleSales uses Oracle One-to-One Fulfillment to process single requests by e-mail, fax, and print output channels. The content of the correspondence is stored in Oracle Marketing Encyclopedia System as reusable master documents. The master documents can be personalized using merge fields and queries. Collateral items (also called deliverables) are uploaded using Oracle Marketing.

Oracle TeleSales generates a fulfillment request when an agent chooses to send a master document or an electronic collateral item. Oracle TeleSales also records the status of the fulfillment request in Oracle Customer Interaction History.

For information about Oracle Collections, see *Oracle TeleSales Implementation Guide* and *Oracle TeleSales User Guide*.

Oracle TeleService Oracle TeleService is a problem tracking application. Agents use Oracle TeleService to manage service requests. Agents also use Oracle TeleService to send correspondence, such as a status, to a party.

Oracle TeleService uses Oracle One-to-One Fulfillment to process single requests by e-mail. The content of the correspondence is stored in Oracle Marketing Encyclopedia System as reusable master documents. The master documents can be personalized using merge fields and queries. Seeded templates are used to join master documents and collateral item in a single package.

Oracle TeleService generates a fulfillment request when an agent chooses to send a template. Oracle TeleService also records the status of the fulfillment request in Oracle Customer Interaction History.

For information about Oracle Collections, see *Oracle TeleService Implementation Guide* and *Oracle TeleService User Guide*.

3.3 Conditional Dependencies

Conditional dependencies provide desired functionality to Oracle One-to-One Fulfillment but are not required for Oracle One-to-One Fulfillment to function.

The following conditional dependencies apply to Oracle Interaction Center:

Oracle Marketing

Oracle Marketing can be used to allow parties to set their contact channels. The Unsubscribe option in the Fulfillment Server configuration causes the Fulfillment Server to insert a hypertext link into the content of e-mail that will allow e-mail recipients to modify their contact preferences.

For e-mail requests, the outgoing e-mail contains an embedded URL to a Web page. The URL leads to the Contact Preferences page in Oracle Marketing Online. The Contact Preferences page allows the recipient to set contact channels.

> **Note:** Oracle Marketing is required to upload collateral items. Collateral may be included in fulfillment request that are created using Oracle Marketing, Oracle Scripting, Oracle Sales Online, and Oracle TeleSales.

3.4 Related Documentation

The latest product documentation is available on Oracle*MetaLink* at http://metalink.oracle.com.

3.5 Installing Oracle One-to-One Fulfillment

You have the following options for installing Oracle One-to-One Fulfillment:

> **Note:** Prior to installation or upgrade, *always* review the readme and patch list for the Oracle Interaction Center Family Pack. The readme contains installation instructions, a summary of product changes, and a list of known issues for a family pack. The patch list contains a list of patches that are required or recommended for use with a family pack. The readme and patch list for the Oracle Interaction Center Family Pack are available on Oracle*MetaLink* at http://metalink.oracle.com. (Perform an advanced search for Document ID 219238.1. Enter the document ID in the Search Field and select the Doc ID option.)

- Oracle Applications Rapid Install

  The Rapid Install is intended for customers who are installing Oracle Applications for the first time or upgrading to Release 11i from to Release 11.0 or Release 10.7. It contains the family packs or product minipacks for all products in Oracle Applications.

  The Rapid Install is provided on CD-ROMs and is available from Oracle Store at http://oraclestore.oracle.com. For information about installing Oracle Applications using Rapid Install, see *Installing Oracle Applications*. For information about upgrading Oracle Applications using Rapid Install, see *Upgrading Oracle Applications*.

- Oracle Applications Maintenance Pack

  The Maintenance Pack is intended for customers who have already installed Oracle Applications Release 11i. It contains the family packs or product minipacks for all products in Oracle Applications.

  The Maintenance Pack is provided as a patch and is available on Oracle*MetaLink* at http://metalink.oracle.com. For information about upgrading Oracle Applications Release 11i using the Maintenance Pack, see Maintenance Pack Release Instructions on Oracle*MetaLink* at http://metalink.oracle.com.

- Oracle Interaction Center Family Pack

  The Oracle Interaction Center Family Pack are intended for customers who have already installed or upgraded to Oracle Applications Release 11i and wish to upgrade only Oracle Interaction Center. The family pack is cumulative and contains only the minipacks for products in the Oracle Interaction Center family.

> **Note:** Oracle Interaction Center products integrate with other products in Oracle Applications. Therefore, you may have to install family packs, product minipacks, or individual product patches for *other* products before installing the Oracle Interaction Center Family Pack.

The Oracle Interaction Center Family Pack is provided as a patch and is available on OracleMetaLink at http://metalink.oracle.com. The readme and patch list for the Oracle Interaction Center Family Pack are available on OracleMetaLink at http://metalink.oracle.com. (Perform an advanced search for Document ID 232834.1. Enter the document ID in the Search Field and select the Doc ID option.)

3.6 Accessing Oracle One-to-One Fulfillment

Oracle Applications uses two different technology stacks:

- Oracle Application Object Library (AOL)
- Java Technology Foundation (JTF)

AOL applications are Oracle Forms developed using Oracle Developer and are usually referred to as Forms-based application. JTF applications are Java Server Pages (JSPs) developed using Oracle JDeveloper and are usually referred to as HTML-based applications. Each type of application accesses the same database and can share information with the other.

The product interfaces are accessed by providing the Uniform Resource Locator (URL) for the environment in an Oracle Applications 11i-compliant Web browser and navigating to the link for the login page for the specific technology stack. You can also provide the URL for the specific login page. This URL is referred to as your login URL.

Oracle Applications URL Use this URL to navigate to the Personal Home Page URL or the CRM Home page URL.

```
http://<host>:<port>/
```

- To navigate to the Personal Home Page URL, choose **Apps Logon Links** > **Personal Home Page**.
- To navigate to the CRM Home Page URL, choose **Apps Logon Links** > **CRM Home Page**.

CRM Home Page URL This URL is sometimes referred to as the Apache or JTF login URL. Use this URL to open the login page for HTML-based applications.

```
http://<host>:<port>/OA_HTML/jtflogin.jsp
```

Personal Home Page URL: This URL is sometimes referred to as the Self-Service Web Applications or SSWA login URL. Use this URL to open the login window for Oracle Applications by the Personal Home Page. You can access Forms-based or HTML-based applications from the Personal Home Page.

```
http://<host>:<port>/OA_HTML/US/ICXINDEX.htm
```

Forms URL Use this URL to open the login page for Forms-based applications. This login URL is typically used by system administrators, not end users.

```
http://<host>:<port>/dev60cgi/f60cgi
```

3.6.1 User Accounts

An application user is an authorized user of Oracle Applications and is uniquely identified by a username. After the user account has been defined, the application user can sign on to Oracle Applications at the CRM Home Page, Personal Home Page, or Forms login.

Note: Oracle Applications is installed with a system-defined username and password.

- Username: sysadmin
- Password: sysadmin

An application user enters a username along with a password to sign on to Oracle Applications. The password assigned by the system administrator is temporary. When signing on for the first time, the application user will be prompted to change the password. Access to specific functionality and data will be determined by the responsibilities assigned to your user account.

3.6.2 Responsibilities

A system administrator assigns one or more responsibilities to an application user. A responsibility is a level of authority that allows a user to access specific functionality and data in Oracle Applications. Oracle Applications is installed with predefined responsibilities. A system administrator can modify a predefined responsibility or create custom responsibilities.

The following table describes the predefined responsibilities that are used to implement Oracle One-to-One Fulfillment.

Responsibility	Function	Interface
CRM HTML Administration	Assign the JTF_ADMIN_FM role to the user account.	HTML
CRM Resource Manager	Create an employee (if Oracle Human Resource Management System is not installed). Import an employee as a resource.	Forms

Responsibility	Function	Interface
One-to-One Fulfillment Administrator	Define Fulfillment Servers and output servers. Create a fulfillment agent group. Associate a fulfillment agent group with a Fulfillment Server. Create queries. Create and/or upload master documents. Create content types and cover letters (content items). View request statuses and history.	HTML
Fulfillment Test Harness	Verify single request and mass request. Create fulfillment APIs and Fulfillment Server setup.	HTML
System Administrator	Create a user for administering Oracle One-to-One Fulfillment. Set values for profile options.	Forms
HRMS Manager, for example US HRMS Manager	Create an employee (Only if Oracle Human Resource Management System is installed).	Forms

In the Forms interface, if an application user has only one responsibility, then the related menu or application (if there is only one function in the menu) appears after the user signs on. If an application user has more than one responsibility, then a list of available responsibilities appears after the user signs on. To switch responsibilities, choose Switch Responsibility from the File menu.

In the HTML interface, an application user must select a default responsibility (even if the user has only one responsibility). The next time the application user signs on, the tabs related to the default responsibility appear. To switch responsibilities, go to Navigation Preferences in your profile (Profile icon). In the Switch Responsibilities section, select another responsibility from the Current Responsibility list.

4 Implementation Tasks

This chapter provides a high-level overview of the implementation task sequence, and references to procedures that you will do during the implementation.

Topics include:

- Section 4.1, "Implementation Task Sequence"
- Section 4.2, "Defining an Administrator"
- Section 4.3, "Defining an Agent"
- Section 4.4, "Configuring Servers"
- Section 4.5, "Implementing a Fulfillment Server"

4.1 Implementation Task Sequence

Implement Oracle One-to-One Fulfillment in the sequence outlined in the following table. The Number column indicates the step order. The Required column indicates whether a step is required. The Description column describes a high-level step and, where applicable, provides a reference to a more detailed topic in this document. The Responsibility column indicates the Oracle Applications user account responsibility that is required to complete the step.

Table 4–1 Sequence of Implementation Procedures

Step	Required	Description	Responsibility
1	Required	**Create an Oracle One-to-One Fulfillment administrator.** Use this user account to perform the remaining steps. See: Section 4.2, "Defining an Administrator"	System Administrator

Table 4–1 Sequence of Implementation Procedures (Cont.)

Step	Required	Description	Responsibility
2	Required	**Define a Fulfillment Server.** See: Section 4.5, "Implementing a Fulfillment Server"	One-to-One Fulfillment Administrator
3	For Collections, Lease Management, Marketing, Scripting, Sales Online, Student System, TeleSales, and TeleService	**Add e-mail output channels.** See: Section 4.5, "Implementing a Fulfillment Server"	One-to-One Fulfillment Administrator
4	For Collections, Student System, TeleSales, and Marketing	**Add fax output channels.** See: Section 4.5, "Implementing a Fulfillment Server"	One-to-One Fulfillment Administrator
5	For Collections, Student Systems, TeleSales, and Marketing	**Add print output channels.** See: Section 4.5, "Implementing a Fulfillment Server"	One-to-One Fulfillment Administrator
6	Required	**Set up profile options.** See: Section 4.5.4, "Fulfillment Profile Options"	System Administrator
7	Required	**Create agent users for the business application.** See the implementation guide of the business application.	Varies
8	Required	**Create a fulfillment agent group.** **1.** Define an agent group. **2.** Associate the group with a Fulfillment Server. **3.** Assign agents. See: Section 8.3, "Managing Fulfillment Agent Groups"	One-to-One Fulfillment Administrator
9	For Marketing, Sales Online, and TeleSales (Optional for Scripting)	**Create collateral items and upload them using Oracle Marketing.** See: *Oracle Marketing User Guide*	Oracle Marketing Administrator
10	For Collections, Lease Management, Scripting, Sales Online, Student Systems, TeleSales, and TeleService	**Create master documents and upload them into Oracle Marketing Encyclopedia System.** See: Section 8.5, "Managing Oracle Marketing Encyclopedia System Master Documents"	One-to-One Fulfillment Administrator

Table 4–1 Sequence of Implementation Procedures (Cont.)

Step	Required	Description	Responsibility
11	For dynamic content in MES master documents	**Create a query in Oracle Marketing Encyclopedia System.** See: Section 8.7, "Managing Oracle Marketing Encyclopedia System Queries"	One-to-One Fulfillment Administrator
12	For Sales Online, TeleService, and Scripting (Survey component)	**Create a template.** ▪ Associate one or more master documents with the fulfillment template ▪ Associate one or more collateral items with the fulfillment template See: Section 8.4, "Managing Oracle Marketing Encyclopedia System Templates"	One-to-One Fulfillment Administrator
13	For dynamic content in OCM templates	**Create a query in Oracle Content Manager.** See:	One-to-One Fulfillment Administrator
14	For RightFax fax and print output channels	**Integrate Captaris RightFax with Oracle One-to-One Fulfillment.** See: Section 5.2, "Integrating with Captaris RightFax"	Not applicable
15		**Upload content blocks and images** See:	
16		**Create a content type** See:	
17		**Create a cover letter and select and upload attachments** See: "Administration Tasks"	
18	For Pasta (Oracle) print output	**Set up Pasta XPDF** See:	
19	Required	**Start the server by using one of the following methods:** ▪ Startup script ▪ Remote command ▪ Oracle Interaction Center Server Manager See: *Oracle Interaction Center Server Manager Implementation Guide* and Section 4.5, "Implementing a Fulfillment Server".	Call Center HTML Admin

Table 4–1 Sequence of Implementation Procedures (Cont.)

Step	Required	Description	Responsibility
20	Optional	**Test the implementation.** See: Section 6, "Verifying the Implementation"	One-to-One Fulfillment Administrator

See Also

- Section 4.2, "Defining an Administrator"
- Section 4.5, "Implementing a Fulfillment Server"

4.2 Defining an Administrator

Do the procedures in the following table to create a user account for administering Oracle One-to-One Fulfillment. The Step column indicates the step order. The Required column indicates whether a step is required. The Description column describes a high-level step and, where applicable, provides a reference to a more detailed topic in this document. The Responsibility column indicates the Oracle Applications user account responsibility required to complete the step.

Step	Required?	Description	Responsibility
1	Yes	**Create an Oracle Applications user account and assign responsibilities.** See: Section 4.2.1, "Creating an Administrator User Account"	System Administration
2	Yes	**Assign the JTF_FM_ADMIN role to the user account.** See: Section 4.2.2, "Assigning a Role to a User Account"	CRM HTML Administration

See Also

- Section 4.1, "Implementation Task Sequence"
- Section 4.5, "Implementing a Fulfillment Server"

4.2.1 Creating an Administrator User Account

Use the following procedure to create a user account for administering Oracle One-to-One Fulfillment.

Login

Self-Service Login URL

Responsibility

System Administrator

Prerequisites

None

Steps

1. In the Navigator window, on the Functions tab, choose **Security** > **User** > **Define**.

   The User window appears.

   Use the following guidelines to define Oracle Applications user names:

   - Use only one word.
   - Use only alphanumeric characters ('A' through 'Z', and '0' through '9').
   - Use only the set of characters that your operating system supports for filenames.

2. In the User Name field, enter the name of the user.

   The password is temporary. When a user signs on to Oracle Applications for the first time, the message "Your password has expired" appears and the user is prompted to set a new password.

   Use the following guidelines to define Oracle Applications passwords:

   - Use at least five characters and no more than 100 characters.
   - Use only alphanumeric characters (A through Z, and 0 through 9).

3. In the Password field, enter the password for the user account and then press **Tab**.

   The cursor remains in the Password field.

4. Enter the password again to verify it.

5. To use this account to submit concurrent programs for Oracle One-to-One Fulfillment, select an employee from the Person field to associate with this user account.

6. In the Responsibilities tab, add the following responsibilities:

   - One-to-One Fulfillment Administrator
   - Fulfillment Test Harness

   After you save the user record, it is not possible to delete an assigned responsibility. Oracle Applications maintain audit data for assigned responsibilities.

   To deactivate an assigned responsibility, set the effective end date (in the Effective Dates - To field) of the assigned responsibility to the current date. To activate an assigned responsibility, clear or reset the effective end date.

7. From the **File** menu, choose **Save**.

   You may close the Users window.

See Also

Section 4.2.2, "Assigning a Role to a User Account"

4.2.2 Assigning a Role to a User Account

The JTF_FM_ADMIN role provides access to the tabs in the Oracle One-to-One Fulfillment administration console. You need the JTF_FM_ADMIN role *in addition to* the One-to-One Fulfillment Administrator responsibility. See Section 4.2.1, "Creating an Administrator User Account" for information about assigning a responsibility to a user account.

Use the following procedure to assign the JTF_FM_ADMIN role to a user account.

Login

HTML Login URL

Responsibility

CRM HTML Administration

Prerequisites

The CRM HTML Administration user must have the following roles:

- JTF_SYSTEM_ADMIN_ROLE, to access user maintenance functions
- JTF_FM_ADMIN, to assign the role to another user

Steps

1. Click the Users tab > Registration sub tab.
2. Click the User Maintenance side panel link.

   The Users page appears.
3. In the Enter Search Criteria fields, select Username, enter the user name, and then click **Go**.

   The matching user names appear.
4. Click a user name link.

   The User Details page appears.
5. Click **Roles**.

   The User-Role Mapping page appears.
6. Move JTF_FM_ADMIN from the Available Roles list to the Assigned Roles list.
7. Click **Update**.

   The User-Role Mapping page refreshes.

See Also

Section 4.2.1, "Creating an Administrator User Account"

4.3 Defining an Agent

To use Oracle One-to-One Fulfillment, agents must be defined as "users" and assigned to a Fulfillment agent group. Use the Forms application to do define an agent in the following sequence:

1. Create an employee.
2. Create an application user, and assign Responsibilities to the user.
3. Create a Resource. A Resource is required to record data in Interaction History.

4.3.1 Creating an Employee

You can create an employee in two ways:

- Section 4.3.1.1, "Creating an Employee in Oracle Human Resource Management Systems"

- Section 4.3.1.2, "Creating an Employee in Resource Manager"

> **Note:** If your system has Oracle Human Resource Management (HRMS), then you must use it create employees. If your system does not have Oracle HRMS, then use the Oracle Resource Manager to create employees.

4.3.1.1 Creating an Employee in Oracle Human Resource Management Systems

Use this procedure to create an employee in Oracle Human Resource Management Systems.

Login

Self-Service Login URL

Responsibility

System Administrator

Prerequisites

Create at least one administrator.

Steps

1. In the Navigator window on the Functions tab, select **People** > **Enter and Maintain**.

   The Find Person window appears.

2. Click **New**.

   The People window appears.

3. In the Last field, type the employee's last name.
4. From the Gender list, select Male or Female.
5. From the Type list, select **Employee**.
6. Other fields may be required, depending on the requirements of your business.
7. Select **File** > **Save**.

4.3.1.2 Creating an Employee in Resource Manager

If Oracle Human Resource Manager is not installed on your system, then use the following procedure to create an employee. This procedure populates the same database tables as the People window in Oracle Human Resource Management Systems.

Login

Self-Service Login URL

Responsibility

System Administrator

Prerequisites

Create at least one administrator.

Steps

1. In the Navigator window on the Functions tab, select **Maintain Employee** > **Employee**.

   The Find Person window appears.

2. Click **New**.

   The People window appears.

3. In the Last field, type the employee's last name.
4. From the Gender list, select Male or Female.
5. From the Type list, select **Employee**.
6. Other fields may be required, depending on the requirements of your business.
7. Select **File** > **Save**.

See Also

Section 4.3.2, "Creating a User Account for an Agent"

4.3.2 Creating a User Account for an Agent

Use the following procedure to create a user account for agents who will use Oracle Applications.

Login

Self-Service Login URL

Responsibility

System Administrator

Prerequisites

Create an employee. See Section 4.3.1, "Creating an Employee".

Steps

1. In the Navigator window, on the Functions tab, choose **Security** > **User** > **Define**.

   The User window appears.

   Use the following guidelines to define Oracle Applications user names:

   - Use only one word.
   - Use only alphanumeric characters (’A’ through ’Z’, and ’0’ through ’9’).
   - Use only the set of characters that your operating system supports for filenames.

2. In the User Name field, enter the name of the user.

   The password is temporary. When a user signs on to Oracle Applications for the first time, the message “Your password has expired” appears and the user is prompted to set a new password.

   Use the following guidelines to define Oracle Applications passwords:

   - Use at least five characters and no more than 100 characters.
   - Use only alphanumeric characters (A through Z, and 0 through 9).

3. In the Password field, enter the password for the user account and then press **Tab**.

   The cursor remains in the Password field.

4. Enter the password again to verify it.

5. To use this account to submit concurrent programs for Oracle One-to-One Fulfillment, select an employee from the Person field to associate with this user account.

6. In the Responsibilities tab, assign the appropriate responsibilities for the eBusiness Suite Applications that the agent will use. No specific Fulfillment Responsibility is required.

   After you save the user record, it is not possible to delete an assigned responsibility. Oracle Applications maintain audit data for assigned responsibilities.

   To deactivate an assigned responsibility, set the effective end date (in the Effective Dates - To field) of the assigned responsibility to the current date. To activate an assigned responsibility, clear or reset the effective end date.

7. From the **File** menu, choose **Save**.

   You may close the Users window.

4.3.3 Assigning or Importing a Resource for an Employee

A resource is required to record data in Interaction History. Use the following procedure to create a resource for an employee.

Login

Self-Service Login URL

Responsibility

System Administrator

Prerequisites

Create the employee. See Section 4.3.1, "Creating an Employee".

Steps

1. In the Navigator window on the Functions tab, select **Maintain Resources** > **Import Resources**.

   The Selection Criteria window appears.

2. From the Resource Category list, select **Employee**.

3. Enter any available selection criteria, such as the name of the employee.

4. Click **Search**.

   The Search Results lists the names of all employees who match the search criteria.

5. Clear the Select option of any listed employees for whom you do not want to create a resource.
6. Click Create **Resource**.

   The Default Values window appears. Optionally, you can add or modify this information later.
7. To accept the defaults, click **OK**.

   The Selected Resources window appears. The Comments field indicates whether the resource is a new record, a duplicate record, or a duplicate record with a new role definition. The Select options are selected automatically.
8. Clear the Select options of any employees whom you do not want to save as a resource.
9. Click **Save Resources**.

   A transaction number appears in the Transaction Number field. The transaction number is associated with each resource that was created during this transaction. More than one resource can have the same transaction number.
10. To view the details about a resource, select the resource and then click **Details**.

    The Resource window appears.
11. Verify the following information:
    - The name of the resource in the Name field.
    - The name of the employee in the Source Name field.
    - The user name in the User Name field.
12. Assign any roles and groups that the eBusiness Application may require. Fulfillment agents do not require specific roles and groups.

4.4 Configuring Servers

Use the Server sub tab to configure Fulfillment Servers and output servers.

Tasks

You can perform the following tasks:

- Section 4.4.1, "Creating a Fulfillment Server"
- Section 4.4.2, "Defining an Email Server"

- Section 4.4.3, "Defining a Fax Server"
- Section 4.4.4, "Adding an Available Print Server"
- Section 4.4.5, "Creating a New Print Server for RightFax Printing"

Concepts

- Fulfillment Server
- E-Mail Tracking
- Fulfillment Channels

4.4.1 Creating a Fulfillment Server

Use this procedure to create and configure a Fulfillment Server.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Install and implement a fulfillment output server.

Steps

1. Select the Server tab.

   The Servers page appears.

2. Click **Create**.

   The Server Details page appears.

3. Enter the details about the server.

   a. In the Server Name field, type a name for the server.

      The name will appear in the list of servers on the Servers page. You cannot change this field after the server is saved.

   b. Optionally, from the Start Time lists, select the hour and minute for the Fulfillment Server to start polling the fulfillment request queues in the database. It does not physically start the server. If running, the server will continue to process retrieved fulfillment requests even when it is not polling the database.

c. Optionally, from the Shutdown Time lists, select the hour and minute of the time at which the Fulfillment Server stops polling the fulfillment request queues in the database. It does not physically stop the server. If running, the server will continue to process retrieved fulfillment requests even when it is not polling the database.

d. Optionally, in the Description field, enter a description of the server.

   This field is for informational purposes only.

4. In the Interaction History group, select the outcome and result that will be recorded in Oracle Interaction History for fulfillment requests processed by the server.

   a. From the Request Outcome list, select the outcome that you want to associated with a fulfillment request that has been retrieved from the request queue by the Fulfillment Server.

   b. From the Request Success Result list, select the result that you want to associate with a fulfillment request that has been successfully processed by the Fulfillment Server.

   c. From the Request Failure Result list, select the result that you want to associate with a fulfillment request that has not been successfully processed by the Fulfillment Server.

5. Click **Continue**.

   The server is saved and the Email Servers page appears.

6. To define an e-mail server, see Defining an Email Server.

4.4.2 Defining an Email Server

To send fulfillment items by e-mail, set the e-mail server profiles for the Fulfillment Server and assign one or more e-mail servers to the Fulfillment Server.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Set the e-mail server profiles for the Fulfillment Server.

Steps

1. If you want the Fulfillment Server to insert a hypertext link into the e-mail content that will allow recipients to access their contact preferences in Oracle Marketing, then select **Unsubscribe**.
2. Enter the Unsubscribe text and URL. This text and link will appear at the end of the e-mail message.
3. If you want the Fulfillment Server to track statistics for e-mail requests, then select **Track E-mail**.
4. Enter the Email Tracking URL. This is the URL of the JSP that tracks opened e-mail messages.
5. In the Return Address field, type the "reply to" e-mail address that will be used in the e-mail messages sent by this e-mail server.

   The business application can override this address by providing a reply e-mail address, otherwise the Fulfillment Server will use this address. If the recipient replies to the e-mail, then the reply will be sent to this address.
6. In the From Address field, type the originating e-mail address for e-mail processed using the Fulfillment Server.

   The business application can override this address by providing an originating e-mail address, otherwise the Fulfillment Server will use this address. This address will appear in the From field in the e-mail.
7. In the Summary of E-mail Server area, click **Add E-mail Server**.

   The Create Email Server page appears.
8. In the Email Server Name field, type a descriptive name for the e-mail server.

   This name will appear in the list of e-mail servers on the E-mail Servers page. You cannot change this field after the definition is saved.
9. In the Outgoing Mail Server field, type the name of the SMTP server.

   This is the host name of the SMTP server.

10. In the Port Number field, type the port number that the Fulfillment Server will use to communicate with the e-mail server.

11. Optionally, in the Description field, type a description of the e-mail server.

    This field is for informational purposes only.

12. To enable the bounce back e-mail feature, in the Bounceback Management group, enter the following parameters:

    a. Enable Bounceback: Select to enable bounce back processing.

    b. IMAP Host: Type the IMAP Host Name.

    c. IMAP Port: Type the IMAP Port Number.

    d. Account Name: Type the name of the bounce back e-mail account.

    e. Account Password: Type the bounce back account password.

    f. Run Frequency: Type the frequency with which the process should repeatedly check for bounce back messages in the e-mail account, and then from the list select the frequency in Hours or Minutes.

    g. Return Path: Type the e-mail address to send the bounce back notices. The Return Path overrides the default return path.

13. Click **Continue**.

    The Fax Servers page appears.

14. To define a fax server, see Defining a Fax Server.

4.4.3 Defining a Fax Server

To send fulfillment items by fax, assign one or more fax servers to the Fulfillment Server.

> **Note:** Oracle One-to-One Fulfillment supports only Captaris RightFax for fax server output.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Set the e-mail server profiles for the Fulfillment Server.

Steps

1. Click **Add Fax Server**.

   The Create Fax Server page appears.

2. In the Fax Server Name field, type a descriptive name for the fax server (for example, Fax 4th Fl Room 1234).

   This name will appear in the list of fax servers on the Fax Servers page. You cannot change this field after the definition is saved.

3. In the Fax Server field, type the server name (for example, //faxserver.domainname.com).

4. In the Prefix field, type the prefix number that must be dialed before the fax number is dialed (for example, 9).

5. In the Return Fax Number field, type the fax number that should be used for return faxes.

6. Optionally, in the Description field, type a description of the fax server

   This field is for informational purposes only.

7. Click **Continue**.

   The Printers page appears.

8. To send fulfillment items by printer, assign one or more print servers to the Fulfillment Server. Do one of the following:

   - To add an available print server, see Adding an Available Print Server.
   - To create a new print server, see Creating a New Print Server for RightFax Printing.

4.4.4 Adding an Available Print Server

Use the following procedure to add an existing print server to the Fulfillment Server.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Set the e-mail server profiles for the Fulfillment Server.

Steps

1. In the Print Servers page, from the Available Print Servers list, select a print server.
2. Click **Add Print Server**.

   The page refreshes. The selected print server appears in the list of print servers.
3. Click **Update** to save.
4. Click **Done**.

4.4.5 Creating a New Print Server for RightFax Printing

Use the following procedure to create a new print server and add it to the Fulfillment Server.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Set the e-mail server profiles for the Fulfillment Server.

Steps

1. In the Print Servers page, click **Create New Print Server**.

   The Create Print Server page appears.
2. In the Print Server Name field, type a descriptive name for the print server (for example, RightFax Server).

This name will appear in the list of print servers on the Printers page. It can be different from the actual printer name or alias. You cannot change this field after the definition is saved.

3. In the Connection String field, for a RightFax Print Server, enter the RightFax server's URL.
4. From the Server Type list, select **RightFax**.

> **Note:** For printing with Oracle Marketing Online, see

5. Click **Apply**.

   The Print Servers page appears.

4.5 Implementing a Fulfillment Server

Use the following procedures to implement a server for Oracle One-to-One Fulfillment.

- Section 4.5.1, "Creating a Startup Script for the Fulfillment Server"
- Section 4.5.2, "Creating a Remote Command Script for the Fulfillment Server"
- Section 4.5.3, "Integrating the Fulfillment Server with Interaction Center Server Manager"
- Section 4.5.4, "Fulfillment Profile Options"

4.5.1 Creating a Startup Script for the Fulfillment Server

Do the steps in the following table to create a startup script for controlling the operation of a Fulfillment Server. The Step column indicates the step order. The Required column indicates whether a step is required. The Description column describes a high-level step and, where applicable, provides a reference to a more detailed topic in this document. The Responsibility column indicates the Oracle Applications user account responsibility that is required to complete the step.

> **Note:** If the Interaction Center Server Manager will be used to monitor and run the Fulfillment Server, then a startup script is not required.

Table 4–2 Steps to Create a Fulfillment Server Startup Script

Step	Required	Description	Responsibility
1	Yes	**Create a script for starting the Fulfillment Server process.** See: Section 4.5.1, "Creating a Startup Script for the Fulfillment Server"	Not applicable
2	No	**Create a script for controlling a Fulfillment Server that is running the background.** See: Section 4.5.2, "Creating a Remote Command Script for the Fulfillment Server"	Not applicable

> **Note:** The installation of Oracle Applications creates a startup script. The default server identifier is 5000. You can manually edit the startup script.

Use the following procedure to create a script to start the Fulfillment Server.

Login

Not applicable

Responsibility

Not Applicable

Prerequisites

- Configure a Fulfillment Server in the Oracle One-to-One Fulfillment administration console.
- Obtain the server identifier that was assigned to the Fulfillment Server definition in the administration console. You can find it on the Server tab.
- Obtain the path to the java.exe (JDK) or jre.exe (JRE) file.
- Obtain the path to the database connectivity file.

Steps

1. Use a text editor, such as Microsoft Notepad, to create a text file that will act as the startup script.

Every startup script must have, at a minimum, the following:

- Maximum memory to be allocated by the Java Virtual Machine (JVM)
- Reference to the identifier for the Fulfillment Server configuration
- Reference to the database connectivity file
- Reference to the executed Java class that runs the Fulfillment Server

See Guidelines for more information.

> **Note:** Always check for syntax errors, such as misspellings and extra spaces.

2. If the startup script is for UNIX (a shell script), save the file with a *.sh extension.
3. If the startup script is for DOS (a batch file), save the file with a *.bat extension.
4. Save the script to a computer that has access, by a network, to the Oracle Applications environment, the Oracle Applications database, and, if you have CLASSPATH statements in your script, any Java files referenced in those statements.

   See Section 3.1.1.1, "Java Files" for information about necessary Java files and the CLASSPATH parameter.
5. To start the Fulfillment Server, change to the directory that contains the script and run the script.

   See Appendix B, "Fulfillment Server Commands" for information about operating the Fulfillment Server from the same command line from which it was started.

   See Section 4.5.2, "Creating a Remote Command Script for the Fulfillment Server" for information about operating the Fulfillment Server from any command line on the computer that is hosting the Fulfillment Server.

Guidelines

The following code example is a template for a basic UNIX shell script for starting a Fulfillment Server.

Note: Do not insert a carriage return at the end of the exec command in the script. You can put the parameters on one line or use a back slash (\) at the end of each line to ensure that the parameters are read as a continuous line.

```
#!/bin/sh

exec <full_path_to_java_or_jre_executable> -mx64m \
-Dengine.ServerID=<server_identifier> \
-Dengine.AOLJ.config=<full_path_to_dbc_file> \
oracle.apps.jtf.fm.FulfillmentServer
```

The following is a template for a basic DOS batch file for starting a Fulfillment Server.

```
<full_path_to_java_or_jre_executable> -mx64m
-Dengine.ServerID=<server_identifier>
-Dengine.AOLJ.config=<full_path_to_dbc_file>
oracle.apps.jtf.fm.FulfillmentServer
```

The -mx parameter sets the maximum memory that the Java Virtual Machine will allocate to the Fulfillment Server. The recommended maximum allocation is 64 megabytes or higher. If you get an OutofMemoryException, then you should increase the maximum allocation. The -ms parameter, which is optional for the Fulfillment Server, sets the minimum memory allocation.

Note: If you are using jre.exe (JRE), then add the -nojit option to the script. For example, in UNIX,

```
exec <full_path_to_java_or_jre_executable> -mx64m -nojit \
```

This option turns off the "just in time" compiler and allows logged errors to display error line numbers.

The engine.ServerID parameter identifies the configuration that is used when the server is started by the script. A server identifier is assigned when the server configuration (also referred to as a server) is created in the Oracle One-to-One Fulfillment administrative console. Servers and their identifiers are listed on the Server tab.

The engine.AOLJ.config parameters identifies the full path to the database connectivity file. The database connectivity file contains the connection parameters for the Oracle Applications database. See Section 3.1.1.3, "Database Connectivity File" for more information.

The executed oracle.apps.jtf.fm.FulfillmentServer class is the main class used to run the Fulfillment Server. The class resides in the apps.zip file. The Fulfillment Server runs on the Admin server and not the Web server.

Note: The oracle.apps.jtf.fm.FulfillmentServer class is a wrapper class around the oracle.apps.jtf.fm.engine.processor.Processor class. It adds extra debug capability. Oracle recommends that you use the oracle.apps.jtf.fm.FulfillmentServer class. However, the server will run with only the oracle.apps.jtf.fm.engine.processor.Processor class.

You can include additional parameters in the startup script. See Appendix A, "Fulfillment Server Commands and Startup Scripts" for more information.

See Also

- Section 4.1, "Implementation Task Sequence"
- Section 4.2, "Defining an Administrator"
- Section 4.5.2, "Creating a Remote Command Script for the Fulfillment Server"

4.5.2 Creating a Remote Command Script for the Fulfillment Server

The remote command script allows you to interact with the Fulfillment Server without the need to maintain the original command prompt used to start the server. When you want to issue a command, you type the name of the remote command script and the command.

The Fulfillment Server is configured in the startup script to monitor a port for remote commands. The port is specified in both the startup script for the server and the remote command script. The remote command script runs on the same computer as the Fulfillment Server or on a computer that is on the same network as the Fulfillment Server.

When a command is sent through the remote command script, the computer that issued the command waits for a response and terminates activity after it receives the response. To prevent unnecessary workload for the Fulfillment Server, the computer that issues the remote command only retains a connection to the Fulfillment Server for the duration of the remote command.

Use the following procedure to create a remote command script and operate a running Fulfillment Server by remote command.

Login

Not applicable

Responsibility

Not Applicable

Prerequisites

Identify an available port for sending commands to the Fulfillment Server by the startup script. The port must be specified in the remote command script and the startup script. For assistance, contact your Oracle Applications system administrator or database administrator.

Steps

1. Stop (use the server parameter "kill") the Fulfillment Server if it is running.
2. Use a text editor, such as Microsoft Notepad, to add the engine.CommandPort parameter to the startup script.

   The engine.CommandPort flag is used to identify the port number, on the computer that is hosting the Fulfillment Server, that the Fulfillment Server will monitor for remote commands.

   In a UNIX startup script, add:

   ```
   -Dengine.CommandPort=<port_number> /
   ```

   In a DOS startup script, add:

   ```
   -Dengine.CommandPort=<port_number>
   ```

   See Section 4.5.1, "Creating a Startup Script for the Fulfillment Server" for information about startup scripts.
3. Use a text editor to create a separate text file that will act as the remote command script.

   Every remote command script must have, at a minimum, the following:

- Reference to the executed Java class that runs the Remote Monitor, an internal monitor in the Fulfillment Server
- A port for sending commands to the Fulfillment Server

Optionally, if the remote command script will reside on a computer other than the computer that is hosting the Fulfillment Server, then you must also specify the name of the computer.

See Guidelines for more information.

> **Note:** Always check for syntax errors, such as misspellings and extra spaces.

4. If the remote command script is for UNIX (a shell script), save the file with a *.sh extension.
5. If the remote command script is for DOS (a batch file), save the file with a *.bat extension.
6. Start the Fulfillment Server (run the startup script).
7. At a different command prompt, send a command using the remote command script.

   For example, enter: `<remote_command_script_name> <command>`
8. Verify that the Fulfillment Server received the command.

Guidelines

The following is an example of a UNIX shell script for issuing commands to a Fulfillment Server that is running in the background.

```
#!/bin/csh

java oracle.apps.jtf.fm.engine.remote.RemoteCommand -port <port_number> [-host
<remote_hostname>] $*
```

The following is an example of a DOS batch file for issuing commands to a Fulfillment Server that is running in the background.

```
java oracle.apps.jtf.fm.engine.remote.RemoteCommand -port <port_number> [-host
<remote_hostname>] %*
```

The value of the port parameter must match the value of the engine.CommandPort parameter in the startup script. If the remote command script is on a computer other than the computer that is hosting the Fulfillment Server, then specify the name of the computer in the host parameter.

The remote command also works from a command prompt with the following command:

```
java oracle.apps.jtf.fm.engine.remote.RemoteCommand -port <port_number> [-host
<remote hostname>] <command>
```

See Also

Section 4.5.1, "Creating a Startup Script for the Fulfillment Server"

4.5.3 Integrating the Fulfillment Server with Interaction Center Server Manager

Oracle Interaction Center Server Manager is the Java server process that is required to be installed on each target computer that runs interaction center servers, including Oracle One-to-One Fulfillment.

Oracle Interaction Center Server Manager has the following key benefits and features:

- Self-service and integrated interface for setup, configuration and administration of servers
- Server process control for server startup and shutdown
- Server process status monitoring that includes server up or down status, and view log files

Oracle Interaction Center Server Manager is part of the Oracle Advanced Inbound HTML administration module. Administrators can use Oracle Interaction Center Server Manager to control and monitor the Interaction Center servers, and to make server installation easier by using of Oracle Rapid Installer and by standardizing the server installation processes.

See Also

- Section 5.1, "Integrating With Oracle Interaction Center Server Manager"
- *Oracle Interaction Center Server Manager Implementation Guide*

4.5.4 Fulfillment Profile Options

The following table lists the Oracle One-to-One Fulfillment profile options.

Table 4–3 Oracle One-to-One Fulfillment Profile Options

Release 11.5.9 and Earlier Profile Name	Release 11.5.10 Profile Name	Seeded Description	Description	Example
JTF_FM_ENABLE_GUI_DEBUG	JTO: Debug: Enable UI Debug	When true, write debug output directly to the generated JSP page.	Print debug statements on JSP pages.	<blank>
JTF_FM_TRACK_IMG_URL	JTO: Email: Tracking URL	URL to embed in e-mail messages used to register when a recipient opens a received email. See the Implementation Guide for this value.	Used by the server as template creating e-mail tracking URLs. Tracking Link used by a JSP to register when a recipient opens a received e-mail. Request ID and Party ID supplied by request details and are encrypted.	`<img src="http://<host>:<port>/OA_HTML/jtffmImageTracker.jsp?PAGE.OBJ.requestId=&PAGE.OBJECT.objId=&file_name=jtfutrpx.gif" height="0" width="0" alt="">`
JTF_FM_UNSUBSCRIBE_TEXT	JTO: Email: Unsubscribe URL	The unsubscribe text at the end of the e-mail message and the URL link to allow recipients to add their e-mail addresses to a Do Not Contact list.	Used by the server as a template for creating unsubscribe URLs as a means of being added to a Do Not Contact list for e-mail. Request ID and Party ID supplied by request details and are encrypted.	"To unsubscribe, please click <a href="http://jtotest.us.oracle.com/OA_HTML/amsContactPreferencesMain.jsp?PAGE.OBJ.requestId=&PAGE.OBJ.objId=">here< a/>
JTF_FM_CREATE_DEST	JTO: Integration: OSO Create Page JSP	The JSP filename of the page invoked from the Fulfillment Status page Create button.	Destination of the Create page button.	<blank>
JTF_FM_DONE_DEST	JTO: Integration: OSO Done Page JSP	The JSP filename of the page to return to after submitting a request.	Destination of the Create page button.	jtffmast.jsp

Table 4–3 Oracle One-to-One Fulfillment Profile Options (Cont.)

Release 11.5.9 and Earlier Profile Name	Release 11.5.10 Profile Name	Seeded Description	Description	Example
JTF_FM_IH_ FAILURE_ RESULT_ID	JTO: Interaction History: Failure Result ID	Interaction History result ID assigned to failed requests.	Set the Interaction History result ID of failed dispatches of the Fulfillment Server.	9
JTF_FM_IH_ OUTCOME_ID	JTO: Interaction History: Outcome ID	Interaction History outcome ID assigned to requests.	Set the Outcome code of Interactions logged by the Fulfillment Server.	10
JTF_FM_IH_ SUCCESS_ RESULT_ID	JTO: Interaction History: Success Result ID	Interaction History result ID assigned to successful requests.	Set the Interaction History result id of successful dispatches of the Fulfillment Server.	8
JTF_FM_ LMERGE_ DELIM	JTO: Merge Delimiter: Left	Merge field left delimiter.	Designate the left delimiter of merge fields.	<< or {(
JTF_FM_ LENCRYPT_ DELIM	JTO: Merge Delimiter: Left Encrypted	Used to designate the left delimiter of encrypted merge fields.	Designate the left delimiter of encrypted merge fields.	<< or {(
JTF_FM_ RMERGE_ DELIM	JTO: Merge Delimiter: Right	Merge field right delimiter.	Designate the right delimiter of merge fields.	>> or)}
JTF_FM_ RENCRYPT_ DELIM	JTO: Merge Delimiter: Right: Encrypted	Designate the right delimiter of encrypted merge fields.	Designate the right delimiter of encrypted merge fields.	>> or)}
JTF_FM_ BOUNCEBACK_ KEY	JTO: Security: IMAP Password Encryption Key	The value to encrypt the bounceback account password.	Key used to encrypt and decrypt bounceback password.	changeme
JTF_FM_ SECURITY_KEY	JTO: Security: Merge Field Encryption Key	The value to encrypt and decrypt merge fields.	Key used to encrypt and decrypt encrypted merge fields.	changeme

Table 4–3 Oracle One-to-One Fulfillment Profile Options (Cont.)

Release 11.5.9 and Earlier Profile Name	Release 11.5.10 Profile Name	Seeded Description	Description	Example
JTF_FM_ DEFAULT_ SERVER	JTO: Server: Default Server ID	The default server to use when the user is not associated with any server.	Used by the APIs. The default server is assumed as the Rapid Install server (serverId - 5000). Groups can set up their own default server in this profile.	5000
JTF_FM_ ERROR_LOG	JTO: Server: Error Log Filename	The server error log filename.	Default Fulfillment Server Error Log name.	Errors.log
JTF_FM_ EVENTS_LOG	JTO: Server: Event Log Filename	Events log file name.	Default Fulfillment Server Event Log name.	Errors.log
JTF_FM_LOG_ LEVEL	JTO: Server: Log Level	Level of logging.	Level of logging used by the Fulfillment Server.	2
JTF_FM_ REFRESH_RATE	JTO: Server: Refresh Idle Milliseconds	The time, in milliseconds, during which the Fulfillment Server rests between self-diagnostic inspections.	Time, in milliseconds, during which the Fulfillment Server rests between self-diagnostic inspections.	20000
JTF_FM_TEMP_ DIR	JTO: Server: Temp Directory	Fulfillment temp directory.	Default location for creating Fulfillment Server temporary files.	tmp

5 Integration Tasks

The following sections explain how to integrate Oracle One-to-One Fulfillment with other Oracle products and third-party products.

- Section 5.1, "Integrating With Oracle Interaction Center Server Manager"
- Section 5.2, "Integrating with Captaris RightFax"

5.1 Integrating With Oracle Interaction Center Server Manager

Oracle Interaction Center Server Manager is part of the Oracle Advanced Inbound HTML administration module. The Oracle Interaction Center Server Manager graphical interface is an efficient way to start, stop and manage all interaction center servers, including the Oracle One-to-One Fulfillment server. This section explains what Oracle Interaction Center Server Manager is and how to integrate it with Oracle One-to-One Fulfillment.

Oracle Interaction Center Server Manager is a Java server process that is installed on each target machine that runs interaction center servers. You can use Oracle Interaction Center Server Manager to do the following Oracle One-to-One Fulfillment processes:

- Set up the Oracle One-to-One Fulfillment Server
- Start and stop the Fulfillment Server
- See the status of the Fulfillment Server
- Inspect the Fulfillment Server Event and Error logs

Use the following procedures to integrate Oracle One-to-One Fulfillment with Oracle Interaction Center Server Manager.

- Section 5.1.1, "Verifying Prerequisites"

- Section 5.1.2, "Starting a Node"
- Section 5.1.3, "Creating a Server Group"
- Section 5.1.4, "Creating and Adding a Fulfillment Server to a Server Group"
- Section 5.1.5, "Configuring Fulfillment Server Parameters"
- Section 5.1.6, "Configuring Server Arguments"
- Section 5.1.7, "Viewing the Status of a Node"
- Section 5.1.8, "Specifying an IP Address for an Interaction Center Server"
- Section 5.1.9, "Starting the Fulfillment Server"
- Section 5.1.10, "Viewing a Log File for the Fulfillment Server"
- Section 5.1.11, "Stopping a Fulfillment Server"
- Section 5.1.12, "Starting the Fulfillment Server in the Command Prompt Console and in Oracle Interaction Center Server Manager"

5.1.1 Verifying Prerequisites

Be sure that the following prerequisites for installing and implementing Oracle Interaction Center Server Manager have been completed before setting up the Fulfillment Server. For details on these procedures, see *Oracle Interaction Center Server Manager Implementation Guide*.

- Apply the following patches:
  - JTO: 3124744 (11.5.10 patch)
  - ICSM: 2840102 and 3069578
- Log in as the Fulfillment Administrator, and define the appropriate servers:
  - One Fulfillment Server
  - One or more e-mail servers (if the e-mail channel is to be used)
  - One or more fax servers (if the fax channel is to be used)
  - One or more print servers. (if the print channel is to be used)
- Install the appropriate Interaction Center Family Pack. It includes Oracle Interaction Center Server Manager.
- Include a fully-patched version of CRM Resource Manager in the environment.

- Assign the Call Center HTML Administration Responsibility to the One-to-One Fulfillment Administrator.
- Log in as the One-to-One Fulfillment Administrator, and then select the Call Center HTML Administration Responsibility.
- Configure a node by completing the steps described in the Oracle Interaction Center ICSM tab > Setup sub tab. A node is a computer that has Oracle Interaction Center Server Manager installed and running.

Note: A node must have a static IP address.

- Start the Oracle Interaction Center Server Manager node before starting the Fulfillment Server within Oracle Interaction Center Server Manager. See Section 5.1.2, "Starting a Node".

5.1.2 Starting a Node

Start an Interaction Center Server Manager node at the command line or, if registered as a Windows NT Service, as a service.

Use the following procedure to start an Interaction Center Server Manager node.

Login

Not Applicable

Responsibility

Not Applicable

Prerequisites

None

Steps

1. Open a command prompt window.

   For example, in Windows NT:

   a. Choose **Start** > **Run**.

   b. Enter **cmd**.

   c. Click **OK**.

2. Change to the scripts directory.
3. Start Interaction Center Server Manager.

- To start Interaction Center Server Manager at the command line, enter:

  ```
  ieoicsm start
  ```

- To start Interaction Center Server Manager process in console mode (Windows NT only) at the command line, enter:

  ```
  ieoicsm console_start
  ```

- To start Interaction Center Server Manager in the Windows NT Services window:

a. Choose **Start** > **Settings** > **Control Panel**.

b. Double-click **Services**.

   The Services window appears.

c. Click **Oracle ICSM <machinename> <dbname>**.

d. Click **Start**.

   Windows NT attempts to start the service.

e. Click **Close**.

Guidelines

The following code is an example of the output from the command prompt in Windows NT:

```
C:> cd D:\icsm\admin\scripts

D:\icsm\admin\scripts> ieoicsm.cmd start

D:\icsm\admin\scripts>echo off
***************************
You are running ieoicsm.cmd
***************************
Tue 10/23/2001
10:03p
ieo_home = ..\..
"Starting Oracle ICSM"
The Oracle ICSM stcsmith.us.oracle.com service is starting.
The Oracle ICSM stcsmith.us.oracle.com service was started successfully.
```

```
ieoicsm.cmd exiting with status 0
```

5.1.3 Creating a Server Group

When the Oracle Interaction Center Server Manager node is running, the next step in integrating Oracle One-to-One Fulfillment is to create a server group. The server group may consist of only the Fulfillment Server, or it may include other interaction center servers. For the purposes of implementing Oracle One-to-One Fulfillment, these instructions explain how to create a server group that has only a Fulfillment Server. To see details on creating a server group with additional interaction center servers, see *Oracle Interaction Center Server Manager Implementation Guide*.

Use the following procedure to create a server group.

Login

HTML Login URL

Responsibility

Call Center HTML Administration

Prerequisites

- Create a Fulfillment Server. See Section 4.4.1, "Creating a Fulfillment Server".
- Verify the prerequisites for installing and implementing Oracle Interaction Center Server Manager. See Section 5.1.1, "Verifying Prerequisites"
- Start the Oracle Interaction Center Server Manager node.

Steps

1. Select the ICSM tab > Server Groups sub tab.

   The Server Group List page appears.

2. Click **Create**.

   The Server Group Details page appears.

3. Enter details about the server group.

   a. Enter a unique name for the server group.

   b. Optionally, in the Location field, enter the location of the server group.

      This field is for informational purposes only.

c. Optionally, in the Description field, enter a description of the server group.

   This field is for informational purposes only.

d. Leave the Super Group field blank.

   The super group identifies the global server group for a multi-site interaction center.

e. In the Default Node field, select the IP address of the Fulfillment Server process.

f. In the Server Configuration field, select **None**.

4. Click **Submit**.

   The page refreshes.

5.1.4 Creating and Adding a Fulfillment Server to a Server Group

Use this procedure to add a server process to a server group.

Login

Self-Service Login URL

Responsibility

Call Center HTML Administration

Prerequisites

- Create a Fulfillment Server.
- Create a server group.

Steps

1. Select the ICSM tab > Server Groups sub tab.

   The Server Group List page appears.

2. Click a server group name.

   The Server Group Details page appears.

3. In the Servers area, click **Create**.

   The Server Details - General page appears.

4. Enter the following details about the server.

a. Enter a unique name for the server.

b. In the Type Name field, select **Fulfillment Server**.

c. Optionally, in the Location field, enter the location of the server. This field is for informational purposes only.

d. Optionally, in the Description field, enter a description of the server. This field is for informational purposes only.

e. From the Member Server Group list, select a server group.

   The Using Server Group field is reserved for future use. Leave it blank.

f. In the Primary Node field, select the Interaction Center Server Manager node to which the server process will be assigned.

g. Click **Save**.

   The Server Details - General page refreshes.

5.1.5 Configuring Fulfillment Server Parameters

Use the following procedure to configure Fulfillment Server parameters.

> **Note:** If at any time a user updates any of the Fulfillment Server parameters, then the server must be stopped and restarted for the new parameters settings to take effect.

Login

Self-Service Login URL

Responsibility

Call Center HTML Administration

Prerequisites

Add a Fulfillment Server to a server group.

Steps

1. Select the ICSM tab > Server Groups sub tab.

   The Server Group List page appears.

2. Click the server group name.

   The Server Group Details page appears.

3. In the Servers group, click the Fulfillment Server name.

   The Server Details - General page appears for the Fulfillment Server.

4. Click **Parameters**.

   The Server Details - Parameters page appears.

5. In the Engine Log Path field, enter the name of the ICSM server assigned to the Fulfillment Server that was created previously.

6. In the Server Identifier field, enter the Identifier of the Fulfillment server that you want to start. The ID value is in the Fulfillment Administration Console > Server Summary page.

7. Optionally, in the remaining Parameter Value fields, type or select values for the parameters listed in the following table.

Table 5–1 Fulfillment Server Parameters

Parameter Name	Required	Description	Sample Value
AOLJ Configuration file	Optional	Specifies the full-path location of the dbc file that the server uses to get connections to the database.	/d01/oracle/visjtolapp l/fnd
Command Port	Optional	Specifies the numeric port id, on the server hardware, that the server will listen to for receipt of remote commands. The only other way to send a command to the server is through the command prompt from which it was started (see engine.CommandPromptEnabled). The Default value does not have command port defined.	11.50/secure/visjtotest _visjtol.dbc
Command Prompt Enabled	Optional	Specifies that the server can receive commands through the command prompt from which it was started. When running the process in the background on UNIX or as a service on Windows NT, use the flag Command Prompt.	True (default)

Table 5–1 Fulfillment Server Parameters (Cont.)

Parameter Name	Required	Description	Sample Value
Content History	Deprecated	From Release 11.5.9 onward, this flag is deprecated. Indicates that the Fulfillment Server writes Content Level History, including detailed information about the processing and delivery of the content sent in a Fulfillment Request.	False (default, that is Content History will be written.)
Debug	Optional	Debug at runtime through the use of a debug flag for the Fulfillment Server. The debug information is written to the fulfillment server events log, with debug=log, otherwise it is written to the Central Logging in the database. In some cases through use of third party debugging (for example, Java Mail API), the debug information is written to the console.	-Ddebug=<flag>[,<flag >]. For example: Ddebug=email,request, content, query
Email Text Only	Optional	Specifies that the server is to send text-only e-mail instead of the HTML-based e-mail that it sends for the "DATA" or "QUERY" contents.	False (default)
Engine Log Path	Required	Name of the ICSM server assigned to the Fulfillment Server that was created previously. Set the engine log path to this server name.	ICSM_FULFILLMENT_ SERVER
Error Log	Optional	Specifies the name of the log used by the server to log errors. This parameter can also be defined in the System Profile options.	Errors.log (default)
Event Log	Optional	Specifies the name of the log used by the server to log event details. This parameter can also be defined in the System Profile options.	Events.log (default)
Exit On Failure	Optional	Specifies that the Fulfillment Server is to shut down when the database is unavailable.	False (default)
Fax Enabler	Optional	Indicates the class that the Fulfillment Server is to use for sending faxes. This class must implement the oracle.apps.jtf.fm.engine.disp.FaxEnabler interface to use as a Fax Enabler.	oracle.apps.jtf.fm.engin e.rightfax.RfFaxEnabler Impl

Table 5–1 Fulfillment Server Parameters (Cont.)

Parameter Name	Required	Description	Sample Value
Ident Package	Optional	Specifies that the Fulfillment Server should write class file versions of the Fulfillment Server to the events log.	False (default)
Log Level	Optional	Indicates the level of detail that will be written to the Error log. For each number from 1 to "n" the server will write an extra layer of chained exception traces to the errors log. The exceptions are chained together so that you can trace the source of the errors as they occur.	
Log Requests	Optional	Boolean flag indicating that the original XML-based request pulled from an Advanced Queue by the server is to be written to disk. The requests will be saved as an XML file in the form of #####.xml, where '#####' is the request id. Saves requests in the "requests" directory under the Log Path.	False (default)
No Interactions	Optional	Indicates the Fulfillment Server is NOT to write Interaction History. This is detailed information about the interaction between the agent and each party in the Fulfillment request. This flag should ONLY be used when Interaction History is not needed but a performance boost is needed.	False (default, that is, Interaction History will be recorded).
Prefers Multipart Alternative	Optional	Specifies that the server is to allow a text-only alternative for the DATA or QUERY content that is sent by e-mail.	False (default)
Print Enabler	Optional	Indicates the class that the Fulfillment Server is to use for sending print jobs to the printer. This class must implement the 'oracle.apps.jtf.fm.engine.disp.PrintEnabler interface to be used as a Print Enabler.	oracle.apps.jtf.fm.engine.rightfax.RfPrintEnablerImpl
Refresh Rate (milliseconds)	Optional	Indicates the amount of time between processor refreshes when the server processor checks the status of all sub-processes to determine if they are functioning correctly.	

Table 5–1 Fulfillment Server Parameters (Cont.)

Parameter Name	Required	Description	Sample Value
Result Set Warning Limit	Optional	Indicates how large a ResetSet can be returned from a database query before a warning will be logged to the Fulfillment Server's error log. Warns that an OutOfMemoryError might have been caused by the size of the Result Set.	100 (default)
SMTP Port	Optional	Overrides the default port '25' for e-mail sent to an SMTP e-mail server.	26
SMTP Server	Optional	Overrides the default SMTP server that is set using the Fulfillment Administration Console.	
Server Identifier	Required	Numeric identifier of the Fulfillment Server that you want to start.	10025
Show Missing Files	Optional.	Works in conjunction with the Ident Package parameter and specifies the Fulfillment Server note missing class files in the events log. Useful when wrapping an older version of the Fulfillment Server with the FulfillmentServer class. Works only when the oracle.apps.jtf.fm.FulfillmentServer class starts the Fulfillment Server instead of oracle.apps.jtf.fm.engine.Processor.	False (default)
Show Warnings	Optional	Indicates to display warnings in the error log. By default, warnings do not appear in the error log. If this flag is not set, then the first time a warning appears, a one-time-only message is presented in the events log signifying that a warning was issued.	False (default)

8. Click **Update** to save.
9. Do one of the following:
   - Click **Advanced** to configure advanced settings for this server.
   - Click **Done** to return to the Server Group Details page.

Guidelines

See *Oracle Interaction Center Server Manager Implementation Guide* for details about Interaction Center server parameters.

5.1.6 Configuring Server Arguments

A Server Argument is a flag followed by a value that is written in the following format:

```
-dbc <dbc file location and name>-svr_name <name of the ICSM server relating to
the fulfillment server>
```

The above flag identifies the server name. This flag will be passed to the FmIcsmMain.java at runtime. The class that belongs to the Fulfillment Server will get all the associated parameters based on the name.

For example:

```
-dbc  /d01/oracle/visjto1appl/fnd/11.5.0/secure/jtotest_visjto1.dbc -svr_name
ICSM_TEST_SERVER
```

To see debug in the log files, you can pass the additional parameter `icsmDebug` in the above command, as in the following example:

```
-dbc  /d01/oracle/visjto1appl/fnd/11.5.0/secure/jtotest_visjto1.dbc -svr_name
ICSM_TEST_SERVER  -icsmDebug true
```

Login

Self-Service Login URL

Responsibility

Call Center HTML Administration

Prerequisites

Add a Fulfillment Server to a server group.

Steps

1. Select the Server Groups tab > Server Group Name > Server Name > **Advanced**.

   The Server Details, Advanced page appears.

2. In the Server Arguments field, type the argument for the flag that you want to specify.

3. Click **Done**.

   The Node Details page appears.

5.1.7 Viewing the Status of a Node

Use this procedure to view the status of an Interaction Center Server Manager node.

Login

HTML Login URL

Responsibility

Call Center HTML Administration

Prerequisites

Install and configure Oracle Interaction Center Server Manager.

Steps

1. Select the ICSM tab > Nodes sub tab.

   The Node List page appears. The Node Status column displays the status of each node:

   - A red flag indicates that Interaction Center Server Manager is not running on the target computer.
   - A green flag indicates that Interaction Center Server Manager is running on the target computer.

   Note: If the status flag colors are not obvious, place the mouse cursor over a flag. A tool tip shows "node is up" or "node is down."

2. To view the node details, click **Node Name**.

   The Node Details page appears.

5.1.8 Specifying an IP Address for an Interaction Center Server

An Interaction Center Server Manager node may have more than one IP address.

Use this procedure to select an IP address in the IP address list for a specific server process assigned to an Interaction Center Server Manager node.

Login

HTML Login URL

Responsibility

Call Center HTML Administration

Prerequisites

- Implement Interaction Center Server Manager.
- Add additional IP addresses to the node IP address list.
- Stop Interaction Center Server.

Steps

1. Select the ICSM tab > Nodes sub tab.

   The Node List page appears.

2. Click a Node Name.

   The Node Details page appears.

3. In the IP Address column, do one of the following:

   - Leave the IP Address field blank.

     Interaction Center Server Manager uses *all* of the available IP addresses in the list. If there is only one IP address, then Interaction Center Server Manager uses that IP address.

   - Select an IP address.

     When the server process is running, Interaction Center Server Manager communicates using the selected IP address.

4. Click **Update**.

   The Node Details page refreshes.

5.1.9 Starting the Fulfillment Server

Use this procedure to start the Fulfillment Server process in a server group.

Login

HTML Login URL

Responsibility

Call Center HTML Administration

Prerequisites

- Install and configure Interaction Center Server Manager.
- Create an interaction center server group.
- Assign server processes in the server group to Interaction Center Server Manager nodes.
- Start the Interaction Center Server Manager nodes to which the server processes are assigned.
- Configure Interaction Center Server parameters.

Steps

1. Select the ICSM tab > Server Groups sub tab.

   The Server Group List page appears.

2. Click a server group name.

   The Server Group Details page appears.

3. In the Servers area, in the Stop/Start column, click **Start** for the server process that you want to start.

   The server process could take several minutes to start. Refresh the browser whenever you want to see the current startup status.

   As the server process starts, "Starting" is displayed in the Stop/Start column. The status of each server process is displayed in the Servers area in the Status column.

5.1.10 Viewing a Log File for the Fulfillment Server

Use this procedure to view a log file for a Fulfillment Server. These log files are generated by Oracle Interaction Center Server Manager.

Login

HTML Login URL

Responsibility

Call Center HTML Administration

Prerequisites

Install and configure Oracle Interaction Center Server Manager.

Steps

1. Select the ICSM tab > Nodes sub tab.

   The Node List page appears.

2. Click a Node Name.

   The Node Details page appears.

3. Click the Server Name of a Fulfillment Server.

   The Server Details: General page appears.

4. Click **Advanced**.

   The Server Details: Advanced page appears. The Fulfillment Server logs are listed by File Name. The name of each log file consists of the server name, date of the log, a unique number, and the .log extension. An example of a log file name would be FULFILLMENT_SERVER20031113_151201.log.

> **Note:** Only file names with the extension ".log" are Fulfillment Server log files. Other listed files manage log file properties, but they are not log files. Do not use the following files:
>
> - CENTRAL LOGGING
> - requests
> - contents
> - logprop.sav

5. In the Files group, click the File Name of a log file.

   The log file appears.

5.1.11 Stopping a Fulfillment Server

Use this procedure to stop a Fulfillment Server process on an Interaction Center Server Manager node.

Login

Self-Service Login URL

Responsibility

Call Center HTML Administration

Prerequisites

None

Steps

1. Select the ICSM tab > Nodes sub tab.

   The Node List page appears.

2. Click a Node Name.

   The Node Details page appears.

3. In the Stop/Start column, click **Stop** for the server process that you want to stop. If the Fulfillment Server is not already running, then the Stop button is not present.

   The server process could take several minutes to stop. Refresh the browser whenever you want to view the current shutdown status.

   As the server process shuts down, "Stopping" appears in the Stop/Start column. The status of each server process is displayed in the Servers group in the Status column.

5.1.12 Starting the Fulfillment Server in the Command Prompt Console and in Oracle Interaction Center Server Manager

The following tables compare ways of starting the Fulfillment Server by using the Windows Command Prompt or the Oracle Interaction Center Server Manager graphical user interface.

Executing Commands on a Server That is Running

Command Prompt	ICSM
YES. When a server is running, commands such as "Stat", "kill", and "error" can be typed at the server prompt.	NO. Commands cannot be run through the GUI.

Starting Multiple Instances of the Same Server

Multiple instances are very useful for load balancing. Multiple instances of the same server can help to process a request much faster.

Command Prompt	ICSM
YES. An existing script can be modified and run again to start a second instance. In that case, the first instance would be the primary server and any instance started after would be a secondary server.	YES. Through the ICSM more work is involved in starting multiple instances. For example, to start two instances, first create two ICSM servers that are associated with the Fulfillment Server. Set up all parameters for those servers, and then start both servers.

Viewing Request and Content Logs

Command Prompt	ICSM
YES. When a Fulfillment Server is set to log content and request debug, it creates two directories within the directory in which it writes the event and error logs. For example, if a server has the logging directory set to /tmp, it starts writing event and error logs in /tmp and, if set to log content and debug, it creates two directories, ■ /tmp/content ■ tmp/request and logs every request and content in the respective directory.	NO. If the Fulfillment Server is set up to log content and debug messages, it writes the logs within the "content" and "request" directories. Also, the list of logs (in the ICSM UI) displays both directories, but when a user clicks any one of the directories, the files within them are not displayed. So, the only way to view request and content logs is through the old method.

Executing Remote Commands

Command Prompt	ICSM
YES. By running Java Orale.apps.jtf.fm.engine.remote.RemoteCommand-host fmtest -port 2142	NO. Commands cannot be run through the GUI.

Starting Servers on a Non-Applications Host

Command Prompt	ICSM
YES. To start a server process on a non-applications computer, the following items should be present. ▪ Fulfillment server code ▪ Dependent code (JTT, FND) ▪ Script to start the server ▪ Dbc file of the database When all the above modules are present, start the script as usual.	YES. Do the following: **4.** Run AUTOCONFIG for product IEO. A new ieoicsm.class is generated. **5.** Log in and select the Call Center HTML Administration responsibility. **6.** Select the Setup sub tab, and download ieoicsm.class to the non-applications computer. **7.** At the computer where ICSM is to be installed, use a Web browser to access Oracle Applications. **8.** Log in to the ICSM HTML Administration Page as an Applications user with "Call Center HTML Administration" Responsibility. **9.** Select the ICSM tab > Setup sub tab. **10.** Follow the steps on the Setup page. **11.** For Fulfillment Java servers: `COMMON_TOP/admin/scripts/{instance}/jtffmctl.sh start` The non-applications tier should appear as one of the nodes in ICSM. **12.** Create a server and associate it with the non-applications tier.

5.2 Integrating with Captaris RightFax

Oracle One-to-One Fulfillment supports only the third-party application Captaris RightFax for sending collateral by fax. Use the following procedures to integrate these two applications.

- Section 5.2.1, "High Level Integration Steps"
- Section 5.2.2, "Sending a Fax Fulfillment Request by RightFax"
- Section 5.2.3, "Sending a Print Fulfillment Request by RightFax"

5.2.1 High Level Integration Steps

Integrate Oracle One-to-One Fulfillment with RightFax in the following sequence:

1. Install RightFax Enterprise Server.
2. Install fax board driver from RightFax fax board driver CD-ROM.
3. Install RightFax Integration Module.
4. Install RightFax Oracle CRM Connector.
5. Install RightFax XML Interface.
6. Install RightFax Java API Interface.
7. Configure IIS to create virtual directory to support rfwebcon.dll.
8. Copy RFJavaInt.zip from the RightFax CD-ROM to $JAVA_TOP.
9. Get gluecode from Oracle*MetaLink* note 207491.1.
10. Compile RFFaxEnablerImpl.java (and/or RFPrintEnablerImpl.java).
11. Add RFJavaInt.zip to your classpath.
12. Add the RFFaxEnablerImpl.class (and/or RFPrintEnablerImpl.class) to apps.zip.

5.2.2 Sending a Fax Fulfillment Request by RightFax

The design of the Fulfillment Server allows the user to define a fax solution based on a static API call. This fax solution, written in Java, is called from the Server as specified in the startup script. If the class exists, the Fulfillment Server loads the specified class file and associated class files, with all supporting classes, in the classpath. The Fulfillment Server then matches the class name in the startup property definition -Dengine.FaxEnabler=your.fax.Enabler with the startup script. The class your.fax.Enabler is the user-defined fax solution. To use this class with the Fulfillment Server, the class must meet two requirements:

- Have a default constructor without parameters.
- Implement the interface oracle.apps.jtf.fm.engine.disp.FaxEnabler (available in the apps.zip file).

The oracle.apps.jtf.fm.engine.FaxEnabler interface has one method,

```
public String faxContents(String Sender,
            String SenderFax,
            String SenderPhone,
            String SenderCompany,
            String Recipient,
            String RecipientFax,
            String RecipientPhone,
            String RecipientCompany,
            String Message,
            HtmlContent[] Contents)
  throws EnablerFailedException;
```

- Sender is the person who is sending the fax. This should be defined as an optional field that can take the empty string "" as an argument value.
- SenderFax is the fax number of the person sending the fax. This should be defined as an optional field that can take the empty string "" as an argument value.
- SenderPhone is the phone number of the person sending the fax. This should be defined as an optional field that can take the empty string "" as an argument value.
- SenderCompany is the company of the person sending the fax. This should be defined as an optional field that can take the empty string "" as an argument value.
- Recipient is the person who will be receiving the fax. This should be defined as an optional field that can take the empty string "" as an argument value.
- RecipientFax is the fax number that will be dialed to send the fax. This parameter must have a value.
- RecipientPhone is the phone number of the person who is to receive the fax. This should be defined as an optional field that can take the empty string "" as an argument value.
- RecipientCompany is the company of the person who is to receive the fax. This should be defined as an optional field that can take the empty string "" as an argument value.
- Message is the message that appears on the cover sheet of the fax being sent. This should be defined as an optional field that can take the empty string "" as an argument value.
- Contents are the actual contents to fax.

5.2.2.1 Glue Source Code

See Oracle*MetaLink* Note 207491.1 for the glue code. Insert the glue code into a file called RfFaxEnablerImpl.java. Use the 'javac' command to compile the file into the class file RfFaxEnablerImpl.class. When compiling, make sure that RFJavaInt.zip and apps.zip are in your CLASSPATH. For more information on using javac, see the Sun Microsystems Web site at http://java.sun.com.

5.2.2.2 Implementing the Glue Code for Fax Requests

Using the glue code for sending faxes by RightFax requires adding configuration flags to the Fulfillment Server start script and making sure that RightFax Java APIs and glue code are on the CLASSPATH. The archive file that includes the RightFax APIs, called RFJavaInt.zip, should be included with your RightFax software. The flags to be included in the Fulfillment Server start script are:

- engine.FaxEnabler indicates the class that the Fulfillment Server should use as it's Fax Enabler.
- rightfax.FaxServerURL indicates the intranet location of the RightFax Fax Server.
- fax.TempDir (default is 'tmp') indicates the directory for storing temporary files to create deliverable fax content.

Each flag in the start script must have the form '-Dname=value', for example, `-Dfax.TempDir=c:\temp`. For more information, see Java Properties at the Sun Microsystems Web site at http://java.sun.com.

5.2.3 Sending a Print Fulfillment Request by RightFax

The design of the Fulfillment Server allows the user to define a print solution that is based on a static API call. This print solution, written in Java, will be called from the Server as specified in the startup script. The Server will load the specified class file and associated class files (assuming that the class exists), with all supporting classes, in the classpath. The Server recognizes the class name from the startup property definition '-Dengine.PrintEnabler=your.print.Enabler' in the Server startup script; the class 'your.print.Enabler' being the user-defined print solution. To use with the Fulfillment Server, this class must meet two requirements:

- It must have a default constructor; one that has no parameters.
- It must implement the interface 'oracle.apps.jtf.fm.engine.disp.PrintEnabler' (available in the apps.zip file).

The 'oracle.apps.jtf.fm.engine.PrintEnabler' interface has one method:

```
public String printContents(String PrinterName, HtmlContent[] Contents)
throws EnablerFailedException;

PrinterName is the name/alias of the Printer to send the print job to.
Contents are the actual contents that are to be printed.
```

5.2.3.1 Glue Source Code

See Oracle*MetaLink* Note 207491.1 for the glue code. Insert the glue code into a file called RfPrintEnablerImpl.java, and then use the javac command to compile the file into the class file RfPrintEnablerImpl.class. When compiling, make sure that RFJavaInt.zip and apps.zip are in your CLASSPATH. For more information on using javac, see the Sun Microsystems Web site http://java.sun.com.

5.2.3.2 Implementing the Glue Code for Print Requests

Using the glue code for sending print jobs by RightFax requires adding a couple of configuration flags to the Fulfillment Server start script and making sure that RightFax Java APIs and glue code are on the CLASSPATH. The archive file that includes the RightFax APIs, called RFJavaInt.zip, should be included with your RightFax software. The flags to be included in the Fulfillment Server start script are:

- engine.PrintEnabler indicates the class that the FulfillmentServer should use as it's Print Enabler.
- rightfax.PrintServerURL indicates the intranet location of the RightFax Print Server.
- print.TempDir (default is 'tmp') indicates the directory where temporary files are to be stored for creating deliverable print content.

Each flag should appear in the start script with the form '-Dname=value'. An example of this would be '-Dprint.TempDir=c:\temp'. For more information on this, please go to the http://java.sun.com Web site and look for information on Java Properties.

6 Verifying the Implementation

The purpose of Oracle One-to-One Fulfillment is to fulfill customer requests from other Oracle applications. Therefore, administrators can test the Fulfillment implementation by creating experimental requests in the Fulfillment Test Tool, a separate application from the Fulfillment Administration Console. The Fulfillment Test Tool simulates sending requests from Oracle applications.

> **Note:** The link for Batch Requests is only for use with releases prior to 11.5.10.

Use the following procedures to verify the implementation.

- Section 6.1, "Testing Requests from Oracle Applications"
- Section 6.2, "Testing Requests from Oracle Marketing Online"
- Section 6.3, "Viewing Log Details"

6.1 Testing Requests from Oracle Applications

Use the following procedures to verify the implementation of Oracle One-to-One Fulfillment by sending requests from Oracle applications.

- Section 6.1.1, "Testing Single Requests from Oracle Business Applications"
- Section 6.1.2, "Testing Mass Requests from Oracle Survey"

6.1.1 Testing Single Requests from Oracle Business Applications

A single request is a request from one customer. Create single requests to verify the implementation with the following Oracle business applications: Collections, Student Systems, TeleSales, TeleService, Sales Online, Lease Management, and Scripting Building Block.

Use the following procedure to create a single request.

Login

Self Service Web Applications (SSWA)

Responsibility

Fulfillment Test Harness Tool

Prerequisites

1. Configure at least one Fulfillment Server.
2. Set up at one least agent (user).
3. Create at least one content item (cover letter) or master document.

Steps

1. At the Welcome or Home page, click **Fulfillment Test Harness Tool**.

   The Requests links appear.

2. Click **Single Request**.

   The Single Request Test Tool page appears.

3. In the Test Details group, click the Server ID search icon.

   The Search and Select window appears.

4. In the Search field, enter the Server ID or Server Name and click **Go**. You can use the % wildcard.

   The Search and Select Results window appears.

5. Select the server and click **Select**.

   The window closes and the Test Tool page appears with the selected value in the Server ID field.

6. In the User ID field, type the ID of the agent who initiated the request. To find the ID, go to the Administration Console and select the Group tab > Group Name. In the Group Details page, use the Agents search field to search for the agent’s user name. The Select Agent page appears, listing any corresponding user names and agent IDs.

7. For Student Systems *only* that use Oracle Content Manager, type values in the following fields:
   - Template ID: The ID of the content item (cover letter), for example, 10448.
   - Version ID: This field is filled automatically when you enter a value in the Template ID field.
8. Select or enter the Priority.
9. Optionally, use the search icon to look for values for the Party ID and Party Name fields.
10. If the fulfillment_user_note merge field is in the master document or cover letter, then enter a value in the User Note field.
11. In the Media Type field, select e-mail, fax, print, or a combination of media.

    Note: If Pasta is being used for printing, then there must be an RTF or PDF document already uploaded in the directory.

12. In the Document Details group, from the Content Type list select one of the following types of documents for the document that is being sent.
    - Attachment: Do not use.
    - Collateral: E-mail attachment (static document)
    - Data: Does not contain merge fields
    - Query: Contains merge fields and has an associated query
13. If you use Management Encyclopedia System master documents, then type values in the following fields:
    - Content ID *or* Content Name:
    - Content Quantity: The number of times that the document will be included in the e-mail
14. From the Content Source list, select the name of the content application.
15. If one of the selected Media Types is e-mail, then in the Mail group, type the subject of the e-mail in the Mail Subject field. In the To Address field, type the e-mail address of the recipient.
16. If one of the selected Media Types is Print, then in the Printer field, type the printer network name.

17. If one of the selected Media Types is Fax, then in the fax field, type the fax number.

18. In the Bind Details group, enter details to define a bind variable for a business application.

    a. In the Bind Name field, type a variable, for example, list_id or party_id.

    b. In the Bind Type field, type `number` or `varchar`.

    c. In the Bind Value field, type a parameter value that can be passed and used as part of the business application query, for example, 12345.

19. Optionally, in the Extended Header group, you can add a CC or BCC recipient.

    a. From the Header Name list, select one of the following:

       – To add a CC copy field in the e-mail header, select **`email_cc_address`**.

       – To add a BCC blind copy field in the e-mail header, select **`email_bcc_address`**.

    b. In the Header Value field, type the e-mail address of the CC or BCC recipient.

20. Click **Submit Request**.

6.1.2 Testing Mass Requests from Oracle Survey

A mass request is a request from more than one customer. Create mass requests to verify the implementation of Oracle Survey.

Use the following procedure to create a mass request.

Login

Self Service Web Applications (SSWA)

Responsibility

Fulfillment Test Harness Tool

Prerequisites

Configure at least one Fulfillment Server.

Steps

1. At the Welcome or Home page, click **Fulfillment Test Harness Tool**.

   The Requests links appear.

2. Click **Mass Request**.

   The Mass Request Test Tool page appears.

3. In the Test Details group, click the Server ID search icon.

   The Search and Select window appears.

4. In the Search field, enter the Server ID and click **Go**. You can use the % wildcard.

   The Search and Select Results window appears.

5. Select the server and click **Select**.

   The window closes and the Test Tool page appears with the selected value in the Server ID field.

6. In the User ID field, type the ID of the agent who initiated the request.

7. Optionally, type values in the following fields:

   - Template ID: The content item (cover letter) ID, for example 10448.
   - Version ID: This field is filled automatically.
   - Priority:
   - Party ID: Click the search icon to look for this value.
   - Party Name:
   - User Note:

8. In the Media Type field, select one or more types of output.

9. In the Document Details group, in the Content ID field enter

10. Optionally, from the Content Type list select one of the following:

    - Attachment: Do not use.
    - Collateral: E-mail attachment (static document)
    - Data: Does not contain merge fields
    - Query: Contains merge fields and has an associated query

11. Optionally, enter values in the following fields:
    - Content ID
    - Content Name
    - Quantity
    - Query ID: The ID of a query used to determine recipients of e-mail, fax, and printed documents and to populate merge fields.
12. From the Content Source list, select the name of the content directory.
13. If one of the selected Media Types is e-mail, then in the Mail group, type values in the following fields:
    - Mail Subject: Subject of the e-mail message
    - Printer: Name of the printer on the local area network
    - Email: E-mail address of the recipient
    - Fax: Fax number of the recipient
14. To enable Interaction History logging, select this option.
15. Optionally, in the Bind Details group, enter details to define a bind variable for a business application.
    a. In the Bind Name field, type a variable, for example, list_id or party_id.
    b. In the Bind Type field, type `number` or `varchar`.
    c. In the Bind Value field, type a parameter value that can be passed and used as part of the business application query, for example, 12345.
16. Optionally, in the Extended Header group, you can add a CC or BCC recipient.
    a. From the Header Name list, select one of the following:
       – To add a CC copy field in the e-mail header, select **`email_cc_address`**.
       – To add a BCC blind copy field in the e-mail header, select **`email_bcc_address`**.
    b. In the Header Value field, type the e-mail address of the CC or BCC recipient.
    c. Click the Select option.
17. Click **Submit Request**.

18. Make a note of the allocated Request Identifier, which you can use in the History and Log pages to view request details and any messages of failed requests.

6.2 Testing Requests from Oracle Marketing Online

To verify the implementation with Oracle Marketing Online, create Fulfillment APIs in the Fulfillment Test Harness Tool, a separate application from the Fulfillment Administration Console.

Use one of the following procedures.

- Section 6.2.1, "Testing Requests from Release 11.5.9 of Oracle Marketing Online"
- Section 6.2.2, "Testing Requests from Release 11.5.10 of Oracle Marketing Online"

6.2.1 Testing Requests from Release 11.5.9 of Oracle Marketing Online

With Release 11.5.9 or Oracle Marketing Online, Oracle One-to-One Fulfillment supports only the e-mail Media Type.

Use the following procedure to test the implementation by sending e-mail requests from Release 11.5.9 of Oracle Marketing Online.

Login

Self Service Web Applications (SSWA)

Responsibility

Fulfillment Test Harness Tool

Prerequisites

Configure at least one Fulfillment Server.

Steps

1. At the Welcome or Home page, click **Fulfillment Test Harness Tool**.

   The Requests links appear.

2. Click **Create Fulfillment API**.

   The Test Create Fulfillment API page appears.

3. In the Details group, click the Requestor ID search icon. The Requestor ID is the ID of the agent who initiated the request.

   The Search and Select window appears.

4. In the Search field, enter the User ID and click **Go**. You can use the % wildcard.

   The Search and Select Results window appears.

5. Select the User ID and click **Select**.

   The window closes and the Test Tool page appears with the selected value in the User ID field.

6. In the Template ID field, type or search for the ID of the content item (cover letter). The Template ID is the same as the Content Item ID, for example, 10448.

   The page refreshes with values in the Template ID field and the Version ID field.

7. Optionally, type values for business applications data, such as Campaign or Promotion Name, in the following fields:

   a. Object Type

   b. Object ID

   c. Source Code

   d. Source Code ID

8. Optionally, from the Media Type list select **E** for e-mail.

9. In the Content Details group, from the Request Type list select a content option:

   a. Electronic, mass fulfillment of e-mail, fax and print requests

   b. Label, address label printing (to be implemented in a future release)

   c. Physical Collateral, not currently used by business applications

   d. Single, requests to individual customers by e-mail, fax or print

   e. Test, to test e-mail requests while not recording them in Interaction History, and not populating merge fields

10. Optionally, in the Subject field, type the subject of the message. You can enter merge fields in the Subject.

11. Optionally, to record requests in Interaction History, select the Log User IH option.

12. Optionally, in the Stop List field, to allow requests to bypass the do-not-send options, type one of the following values:

    - Unsubscribe: To not insert the Unsubscribe option in e-mail to the customer.
    - Stoplist: To not inquire whether the customer has requested to not receive e-mail. Use this option when the customer has made a specific request to receive information.
    - Both: To select both Unsubscribe and Stoplist.

13. In the Bind Details group, enter details to define a bind variable for a business application.

    a. In the Bind Name field, type a variable, for example, list_id or party_id.

    b. In the Bind Type field, type `number` or `varchar`.

    c. In the Bind Value field, type a parameter value that can be passed and used as part of the business application query, for example, 12345.

14. In the Address Details group, enter information to identify the recipient. Values that you enter here will override values in the query.

    a. In the Party ID field, type the numerical identifier of the recipient.

    b. In the Email Address field, type the e-mail address of the recipient.

    c. In the Fax Address field, type the fax number of the recipient.

    d. In the Printer Address field, type the printer name or number that identifies the printer on your network.

15. Click **Submit Request**.

    The page refreshes and details about the submitted request appear in the Information header at the top of the page. The header includes the Request Submitted number, which corresponds to the Request Identifier in the Status tab of the HTML Administration Console.

16. Make a note of the allocated Request Identifier, which you can use in the History and Log pages to view request details and any messages of failed requests.

6.2.2 Testing Requests from Release 11.5.10 of Oracle Marketing Online

With Release 11.5.10 of Oracle Marketing Online, Oracle One-to-One Fulfillment supports creating renditions of cover letter formats. The formats include HTML, RTF, PDF, and plain text. In addition, all of the available Media Types are valid.

Use the following procedure to test the implementation by sending requests from Release 11.5.10 of Oracle Marketing Online.

Login

Self Service Web Applications (SSWA)

Responsibility

Fulfillment Test Harness Tool

Prerequisites

Configure at least one Fulfillment Server.

Steps

1. At the Welcome or Home page, click **Fulfillment Test Harness Tool**.

   The Requests links appear.

2. Click **Create Fulfillment Rendition API**.

   The Test Create Fulfillment Rendition API page appears.

3. In the Details group, click the Requestor ID search icon. The Requestor ID is the ID of the agent who initiated the request.

   The Search and Select window appears.

4. In the Search field, enter the User ID and click **Go**. You can use the % wildcard.

   The Search and Select Results window appears.

5. Select the User ID and click **Select**.

   The window closes and the Test Tool page appears with the selected value in the User ID field.

6. In the Template ID field, type or search for the ID of the content item (cover letter). The Template ID is the same as the Content Item ID, for example, 10448.

   The page refreshes with values in the Template ID field and the Version ID field.

7. Optionally, type values for business applications data, such as Campaign or Promotion Name, in the following fields:

   a. Object Type

   b. Object ID

   c. Source Code
   d. Source Code ID

8. Optionally, from the Media Type list select the type of request:
   a. E: e-mail only
   b. EF: e-mail and fax
   c. EP: e-mail and print
   d. EPF: e-mail, fax, and print
   e. F: fax only
   f. P: print only
   g. PF: print and fax
9. In the Content Details group, from the Request Type list select a content option:
   a. Electronic, mass fulfillment of e-mail, fax and print requests
   b. Label, address label printing (to be implemented in a future release)
   c. Physical Collateral, not currently used by business applications
   d. Single, requests to individual customers by e-mail, fax or print
   e. Test, to test e-mail requests while not recording them in Interaction History, and not populating merge fields
10. If the Media Type is e-mail, then in the Subject field, type the subject of the message. You can enter merge fields in the Subject.
11. Optionally, to record requests in Interaction History, select the Log User IH option.
12. Optionally, in the Stop List field, to allow requests to bypass the do-not-send options, type one of the following values:
    - Unsubscribe: To not insert the Unsubscribe option in e-mail to the customer.
    - Stoplist: To not inquire whether the customer has requested to not receive e-mail. Use this option when the customer has made a specific request to receive information.
    - Both: To select both Unsubscribe and Stoplist.
13. In the Bind Details group, enter details to define a bind variable for a business application.

   a. In the Bind Name field, enter a variable, for example, list_id or party_id.
   b. In the Bind Value field, type a parameter value that can be passed and used as part of the business application query, for example, 12345.

14. In the Address Details group, enter information to identify the recipient.
   a. In the Part Id field, type the numerical identifier of the recipient.
   b. In the Email Address field, type the e-mail address of the recipient.
   c. In the Fax Address field, type the fax number of the recipient.
   d. In the Printer Address field, type the printer name or number that identifies the printer on your network.

15. Click **Submit Request**.

    The page refreshes and details about the submitted request appear in the Information header at the top of the page. The header includes the Request Submitted number, which corresponds to the Request Identifier in the Status tab of the HTML Administration Console.

16. Make a note of the allocated Request Identifier, which you can use in the History and Log pages to view request details and any messages of failed requests.

6.3 Viewing Log Details

Use the Logs tab to read recorded details about requests.

Login

Self Service Web Applications (SSWA)

Responsibility

Fulfillment Test Harness Tool

Prerequisites

Make at least one fulfillment request in the Test Tool.

Steps

1. Click the Logs tab.

   The Log Details page appears.

2. To list the highest numbered ID first, click **Request ID** twice .

   The Test Details column indicates whether the request passed successfully or failed. If an exception message is listed, then the API call failed. Ensure that the data that was entered correctly in the Test Tool, and re-enter the data if necessary.

7 Diagnostics and Troubleshooting

Use the following information to diagnose and fix issues you might encounter when implementing and using Oracle One-to-One Fulfillment.

- Section 7.1, "Common Error Messages"
- Section 7.2, "Fulfillment Server Error Messages and Debug Flags"
- Section 7.3, "E-Mail Issues"
- Section 7.4, "Frequently Asked Questions"
- Section 7.7, "Troubleshooting the RightFax Integration"

7.1 Common Error Messages

Refer to the following common error messages for instructions on correcting errors.

7.1.1 Out of Memory Error

An out of memory error is received when running JTFFMCTL.SH.

Solution

Reset JTF_FM_MAX_PROCESSES to 9 or less. The server is designed to run a maximum of nine processes.

7.1.2 java.lang.NullpointerException or Blank Screen

If you receive the following exception:

- java.lang.NullpointerException

  or

- You are able to send collateral using the Fulfillment Manager, but the system stops and the monitor displays a blank screen. In Fulfillment Manager, the request shows as completing successfully. The e-mail is sent, but the forms do not return a Collateral Sent message or return to a Java Server Page.

Solution

1. Set the profile JTF_FM_DONE_DEST to a Java Server Page to which you want to return (such as CntctSumryMain.jsp).
2. Restart the Apache server.

7.2 Fulfillment Server Error Messages and Debug Flags

Refer to the following topics for descriptions of Fulfillment Server error messages and debug flags.

- Section 7.2.1, "Common Fulfillment Server Errors"
- Section 7.2.2, "Fulfillment Server Debug Flags"
- Section 7.2.3, "Stopping and Restarting the Fulfillment Server"

7.2.1 Common Fulfillment Server Errors

The following table lists errors that the Fulfillment server most commonly writes to the logs. For further information regarding each error, see the exception trace files.

Table 7-1 Fulfillment Server Errors

Error	Description
tried to execute bind query >	Unable to execute Bind Query
no columns returned in result set --	No records returned from query
unable to create temp file for	Could not create a temporary file on the local file system
no bind value for merge field	Merge field exists for which there is no matching bind (replacement text)
unable to get form fields	Unable to determine the form fields in a PDF document
unable to merge PDF with XFDF	Merge of bind data with PDF document failed

Table 7-1 Fulfillment Server Errors (Cont.)

Error	Description
multiple HTML files in zipped content	Correct format is one HTML file and one or more image files (gif, jpeg, and so on)
maximum number of images has been reached for content	Memory constraints limit the maximum number of images that can be associated with an HTML file
no HTML in zipped content --	Correct format is one HTML file and one or more image files (gif, jpeg, and so on)
no images in zipped content	Correct format is one HTML file and one or more image files (gif, jpeg, and so on)
could not read any zip file entries	Invalid Zip format
could not process zip file	Invalid Zip format
unable to download with query id	No query associated with the content
invalid extended_header format	Invalid extended header XML format
empty header_name in extended_header	Missing name in name-value pair
unable to create email address from	Invalid string for use as e-mail address
unable to send email -- refer to error	Unable to send an email, refers to an error in the error log
unable to get text/plain content	Unable to get text content
unable to get content bind	Error in getting bind information associated with a Content item
could not get fnd_file_id for request	fnd_file_id not present for a content item in the XML
unrecognized mime type for dynamic content	Dynamic Content (capable of being merged) can only be of type ['text/html','richtext',pdf]
unable to merge legacy data	Merge failure occurs when trying to review a sent document
unable to download with fnd file id	Data not found in fnd_lobs for the file_id

Table 7-1 Fulfillment Server Errors (Cont.)

Error	Description
unable to create directory	Unable to create a directory in the file system

7.2.2 Fulfillment Server Debug Flags

The Fulfillment Server can be debugged at Runtime by using a debug flag in the Fulfillment Server startup script or in the command line. The debug information is written to the Fulfillment Server events log, with debug=log, or to Central Logging in the database.

In some cases, through use of third-party debugging such as Java Mail API, debug information is written to the console. The correct syntax for such flags is `-Ddebug=<flag>[,<flag>]`. The following table lists individual settings of the Fulfillment Server debug flags.

Table 7–1 Fulfillment Server Debug Flags

Flag	Description
full	Turn on all debug flags (*including 'log' which disables Central Logging).
off	Turn off all debugging.
content	Write request content, as downloaded from the database, to the local disk on which the server is running. The directory is called 'contents' and is under the log directory.
email	Print details of e-mail delivery the Events log and to the console. Not all information printed to the console is recorded in the Events log.
fax	Debug fax delivery.
history	Debug the writing of Fulfillment Request History, Fulfillment Content History, and Interaction History.
log	Debug the creation of local disk log files. *Disables the use of Central Logging.
monitor	Debug the operations of the Fulfillment Server internal monitors. All vital internal processes of the server are handled by these monitors.
print	Debug sending print jobs.
processor	Debug the functions of the server main thread.

Table 7–1 Fulfillment Server Debug Flags (Cont.)

Flag	Description
query	Debug queries executed by the server. These are written to the Events log.
request	Debug requests that are processed by the server. Details of the request are written to the Events log.
header	Debug extended headers. Each is written as part of the e-mail header for e-mail sent by the server. Additional details of the request are written to the header such as request id, party_id, and so on.
jdbc	Debug JDBC calls made by the server. Java communicates with the database through JDBC.
fault	Debug the functioning of the server fault monitor. This provides details of request fault detection and handling.
mass	Debug Mass Request processing.
failure	Debug failures that occur as a normal part of processing a request. The failure is written to the Error log to record a stack trace from where the failure originated. This skews the actual error count because an exception is created to track the stack trace of the failure.
merge	Debug the merge field parsing and insertion of bind variables within dynamic request content (Master Documents and e-mail subject field).
queue	Debug reading and writing to Advanced Queues (AQ). The XML-based requests and responses are written to the Events log.
bind	Debug the use of bind data that is associated with the content of a fulfillment request.
labels	Debug PDF Labels requests.

7.2.3 Stopping and Restarting the Fulfillment Server

Stop and restart the Fulfillment server for changes to take effect to the settings, parameters and flags of the following servers:

- Fulfillment Server
- Email Server
- Print Server

- Fax Server

7.3 E-Mail Issues

Use the following solutions to resolve these known issues with the Fulfillment Server sending e-mail.

7.3.1 java.io.IOException: 500 5.5.1 Command Unrecognized

This type of Java IOException, recorded in the Fulfillment Server's error log, usually means that the e-mail server did not recognize the sent command. This behavior may happen because the e-mail server did not recognize the common SMTP command 'EHLO' but instead used the older 'HELO' command.

Solution

1. Use the Fulfillment Administration Console to configure the Fulfillment Server to send e-mail through an outbound e-mail server that recognizes the 'EHLO' command.
2. Restart the Fulfillment Server.

7.3.2 Characters Appear as '?' in Body of E-Mail

The Fulfillment Server may parse characters or symbols incorrectly for some natural languages, other than English, that are not based on US ASCII. The Fulfillment Server sends e-mail that contains unreadable characters, which appear as question marks. This situation may have the following causes:

- The Java Virtual Machine requires instructions on how to resolve these characters through the operating system.
- The Fulfillment Server startup script contains merge delimiters such as `&laquo;` and `&raquo;`. Previous versions of the Fulfillment Server recognized these merge field delimiters, but the current version does not recognize them.

Solution

- Use the configuration settings listed in Section A.2.5, "Language Character Parsing" in Appendix A, "Fulfillment Server Commands and Startup Scripts".
- Use the following flags in the Fulfillment Server startup script.

```
-Dmerge.LeftDelimiterString="&laquo;" -Dmerge.RightDelimiterString="&raquo;"
```

7.4 Frequently Asked Questions

Refer to the following frequently asked questions for help in resolving issues.

7.4.1 Uploading an RTF Document Fails in Oracle Content Manager

When uploading an RTF document in Oracle Content Manager, the upload fails.

Solution

This failure can occur when you have Microsoft Word open when you click the Submit for Approval button. The solution is to close Word and then click Submit again.

7.4.2 Content Types Not Visible

Why can I not see some content types in the list when I create a content item?

Solution

Only active content types appear in the list. It does not include inactive content types.

7.4.3 Cannot Update Existing Content Type

Why am I unable to update content types that are being used by content items?

Solution

Content types that are being used by content items cannot be updated.

7.4.4 Log Directory Path

The Fulfillment server is writing log files to the ./logs subdirectory instead of ./log. How can I change this path?

Solution

The default directory for error messages and logs is ./log. This directory is determined by the oracle.apps.jtf.fm.engine.logging.logger class in the file apps.zip. If logs are written to a different directory (such as ./logs) instead of ./log, then change the logger.class. Open that class and see how DEF_LOG_PATH and LogPath variables are set. Use the -Dengine.LogPath switch to change the logging default directory. The format is: -Dengine.LogPath=<directory>

7.4.5 TeleSales Java Exception

The Fulfillment log receives a Java exception when trying to send collateral from Telesales. The Telesales popup window indicates that it was successful, however, there is nothing in the history for the collateral, and the following error message appears in the Fulfillment log file.

```
'could not complete process for request '10011'
oracle.apps.jtf.fm.engine.request.WorkNotCompletedException: failed to process
part of request '10011' at java.lang.Throwable.fillInStackTrace(Native Method)
```

Solution

Use the following steps to check your Return Address setup for the Fulfillment server that is in use.

1. Log in using the Fulfillment Responsibility.
2. Select Server tab > server name > **Email Servers**.
3. Check that the Return Address setup is a valid e-mail address to which you want to send all e-mail replies.

7.4.6 Missing 3rdParty.zip

What can I do when configuring the Fulfillment Server, if the file 3rdParty.zip is missing?

Solution

When the application compiles, it creates the file 3rdParty.zip. If this file is missing, you can start the Fulfillment server by appending the classpath with `$FND_TOP/3rdparty/stdalone`.

7.4.7 java.lang.IllegalArgumentException

A java.lang.IllegalArgumentException message occurs while trying to start the FM Server Script. What is the problem?

Solution

This condition happens because of an improperly written Fulfillment Server startup script. The PrintEnabler and the FaxEnabler take the class file name from the script. If the file name is not correct, then the method that receives this classfile name will throw the exception. All the -D flags in the script should be written in either of the following ways:

- The Flags should be separated by a "\" and then a new line should contain the next flag. For example:

```
java -ms32m -mx64m\
-Dengine.LogLevel=9\
-Dengine.LogRequests=true\
-Dpkg.ident=true\
-Dengine.PrintEnabler=oracle.apps.jtf.fm.engine.rightfax.RfPrintEnablerImpl\
-Dengine.FaxEnabler=oracle.apps.jtf.fm.engine.rightfax.RfFaxEnablerImpl\
oracle.apps.jtf.fm.engine.processor.Processor
```

- The Flags should be separated by spaces only. There should be no "\" or a carriage return anywhere in the Java call to the server. For example:

```
java -ms32m -mx64m -Dengine.LogLevel=9  -Dengine.LogRequests=true
-Dpkg.ident=true
-Dengine.PrintEnabler=oracle.apps.jtf.fm.engine.rightfax.RfPrintEnablerImpl
-Dengine.FaxEnabler=oracle.apps.jtf.fm.engine.rightfax.RfFaxEnablerImpl
oracle.apps.jtf.fm.engine.processor.Processor
```

7.4.8 DoNotContactException

A DoNotContactException message occurs while trying to send a request through the Fulfillment Server.

The outcome of the request id FAILURE and the following error appear in the Error.log file:

```
Unable to dispatch 'EMAIL' for request '<request_id>'
oracle.apps.jtf.fm.engine.disp.DoNotContactException: unable to email
```

Solution

When a request is submitted to the Fulfillment server, the server checks the APPS.HZ_CONTACT_PREFERENCES table for the values of "DO_NOT", "EMAIL" or "CONTACT" for PREFERENCE_CODE and CONTACT_TYPE, respectively, to determine if the e-mail should not be sent to the customer. If the values are present in the table for the party_id that is specified in the request, then the request fails.

To change these values, do the following procedure:

1. Run the following query in the database for the party_id that is in use.

```
SELECTdistinct b.contact_type
FROMapps.hz_contact_preferences b,
apps.hz_contact_points c
WHERE
```

```
b.preference_code = 'DO_NOT'
AND
((b.contact_level_table = 'HZ_PARTIES' and b.contact_level_table_id =
&party_id)
OR
(b.contact_level_table = 'HZ_CONTACT_POINTS'
AND c.contact_point_id = b.contact_level_table_id
AND c.owner_table_name = 'HZ_PARTIES'
AND c.owner_table_id = &party_id
AND c.status ='A'))
AND nvl(b.preference_start_date,sysdate -1) < sysdate
AND nvl(b.preference_end_date,sysdate+1) > sysdate
ORDER BY b.contact_type
```

2. The Contact_type would appear as 'EMAIL' or 'CONTACT'. Remove the 'EMAIL'/'CONTACT' restriction or change the 'DO_NOT' restriction.
3. Resend the request.

   It should now succeed.

7.4.9 Commands in Background Mode

How can I send commands to the Fulfillment Server while in background mode?

Solution

Do the following procedure:

1. If the Fulfillment server is running in console mode, shut it down by typing **`kill`**. If the Fulfillment Server is running in background mode, kill the process. To be safe, you might want to do a `ps -ef grep java` to make sure that no other Fulfillment Server processes are running. Also, you need to be able to use a port to send commands through a remote command process. You may need to confirm with the database administrator which port is available.
2. To reference the port you will use to accept commands for the Fulfillment Server, add the following line to the startup script, where 99999 represents the selected port.

   ```
   -Dengine.CommandPort=99999
   ```

3. Create a remote command script that is basically one line. Include the port to monitor in this script.

   For UNIX:

```
#!/bin/csh java oracle.apps.jtf.fm.engine.remote.RemoteCommand-port99999
/host <remote hostname>/ $*
```

For DOS:

```
java oracle.apps.jtf.fm.engine.remote.RemoteCommand-port99999 /host <remote
hostname>/ %*
```

4. Start the server by running the Server script in the console mode (not in background). Open another window and test the remote script by sending commands to it. If you name the remote script RS.sh, type a command like **`RS.sh stat`** or **`RS.sh kill`**. If the server that is running in console mode accepts these commands, everything is set up correctly. You can then type **`RS.sh kill`** to shut down the server.
5. If the remote functionality tested successfully, so you can now run the server in the background. If the Fulfillment Server Script was FS.sh, you could run FS.sh &.
6. When the server is running in the background, you can send commands to it by using your remote script RS.sh.

7.4.10 Viewing Request Processing

How can I see what Fulfillment Server processes are processing requests?

Solution

Type the command **`stat`** at the Fulfillment server command prompt to see what processes are processing requests.

7.4.11 "Initializing Processor" Error

When I try to start the Fulfillment Server, I receive the following error message. What is the problem?

```
INITIALIZING PROCESSOR ...
GETTING A DATABASE CONNECTION FOR SERVER CONFIGURATION ...
Processor cannot be started
oracle.apps.jtf.fm.engine.processor.ProcessorException
oracle.apps.jtf.fm.engine.processor.Processor.
initStoredConfiguration
```

Solution

Check for spelling mistakes and syntax errors in the Fulfillment startup script.

7.4.12 Merge Field Errors

What can I do if I encounter either of the following problems when my master documents and cover letters use merge fields?

- The merge fields do not get merged in e-mail messages, fax messages or printed documents sent by Oracle One-to-One Fulfillment.
- The e-mail messages, fax messages or printed documents show the merge fields surrounded by the "?" character, despite the use of the valid characters "<<" and ">>" when creating the master document or cover letter in the Administration Console.

Solution

This issue occurs with Windows NT 4.00.1381, Service Pack 6 with Internet Explorer 5.5.50. The browser is not able to take the standard ASCII value of the "<<" and ">>" merge field delimiters, and therefore the master document or Oracle Content Manager Template contains the "?" delimiters instead of "<<" and ">>".

Use one of the following workarounds:

- Instead of using the "Create and Upload" Master Document/OCM Template option within the One-to-One Fulfillment Administration Console, use the "Upload" Master Document/OCM Template option. Create the document using Microsoft Word 2000, insert the appropriate merge fields with the "<<" and ">>" special character delimiters and then save the document as a Web Page (for example, an HTML file). Use the Upload Master Document/OCM Template option to upload the file.
- If using the "Create and Upload" Master Document/OCM Template page, ensure that the desktop being used has Windows 2000 or a different platform configuration.

7.4.13 Unable to Use Merge Field Delimiters &laquo; and &raquo;

Prior to Release 11.5.8, some versions were missing the left angle quote &laquo; and the right angle quote &raquo;. The workaround is to insert the delimiter flags in the Fulfillment startup script as,

```
-Dmerge.LeftDelimiterString=”&laquo;”
-Dmerge.RightDelimiterString=”&raquo;”
```

Insert the flags after the Java or JRE command and before the name of the executed class. The class is 'oracle.apps.jtf.fm.engine.processor.Processor', in earlier releases, and 'oracle.apps.jtf.fm.FulfillmentServer' in later releases. The classes are case sensitive.

7.4.14 Sales Online Request Failure

I am unable to send a request (for example, no request appears in the Status page) when the user logs into Oracle Sales Online (OSO) and submits a request using the Fulfillment wizard.

Solution

The agent may not be associated with a Fulfillment server (through a Fulfillment group). Execute the following query:

```
SELECT a.server_id
FROM jtf_fm_groups_all a,
jtf_fm_group_fnd_user b,
jtf_fm_fnd_user_v c
WHERE a.group_id = b.group_id
AND b.user_id = c.user_id
AND b.user_id = ?
AND a.f_deletedflag is null
AND b.f_deletedflag is null
```

1. If the query does not return results, add the user to a Fulfillment group that is associated with the server that the customer is trying to start. If the query returns a value, then perform step 2. If a user is not associated with a Fulfillment group, then the default Fulfillment server (with Server ID = 5000) is used.
2. The cover letter that was sent may not have had a query associated with it. This is an existing constraint of the Fulfillment wizard used by Oracle Sales Online. To resolve the issue, associate a simple query (for example, "select sysdate from dual") to the cover letter and resend the request.

7.4.15 Table Rows and Merge Fields

How can I switch from repeating to non-repeating HTML table rows with merge fields?

Solution

Enhancements for this functionality are available in patches for Releases 11.5.8 and 11.5.9, and are part of the base product for Release 11.5.10 and later.

Table 7–2 Table Row Enhancements in Recent Releases

Code Line	Patch Number	Patch Description
11.5.8	3138990	Repeating row merge field enhancements
11.5.9	3121835	Oracle One-to-One Fulfillment Release 11.5.9.0.3
11.5.10	Not applicable	Included in Oracle One-to-One Fulfillment Release 11.5.10

In previous releases Oracle One-to-One Fulfillment treated all HTML source documents containing merge fields embedded in table rows as repeating rows. For example, if users wanted to create an itemized order form, they could create the HTML source document with an HTML table that represented the itemized list. The HTML table could have merge fields in the following columns:

ITEM_NUMBER, ITEM_DESCRIPTION, QUANTITY, UNIT_PRICE, TOTAL_PRICE

Example HTML Code

```
<table border="1" summary="Order Table">
<tr>
    <th>Item Number</th>
    <th>Item Description</th>
    <th>Quantity</th>
    <th>Unit Price</th>
    <th>Total Price</th>
</tr>
<tr>
    <td>{(ITEM_NUMBER)}</td>
    <td>{(ITEM_DESCRIPTION)}</td>
    <td>{(QUANTITY)}</td>
    <td>{(UNIT_PRICE)}</td>
    <td>{(TOTAL_PRICE)}</td>
</tr>
<table>
```

Users can then use a key, such as ORDER_NUMBER, to associate a SQL query to the document that retrieves multiple rows of data from the database, as in the following example:

```
SELECT ITEM_NUMBER,
       ITEM_DESCRIPTION,
       QUANTITY,
       UNIT_PRICE,
       TOTAL_PRICE,
FROM ORDERS
WHERE ORDER_NUMBER = :my_order_number
```

When Fulfillment processes this document, it parses the HTML source document and identifies a table containing merge fields. As Fulfillment executes the query, it repeats the line containing the merge fields for each row that is returned from the query. In this way, a single HTML table row in the source document is custom expanded to all the results that are returned from the associated query.

Because this feature repeated any HTML table row with merge fields, it prevented some users from using HTML tables for document layout.

Solution

To accommodate users who want to use HTML tables for document layout, and also to support repeatable fields, from Release 11.5.10 onward Fulfillment can create both repeatable merge field rows and non-repeatable merge field rows.

To keep the existing result of every table row containing repeated merge fields, do not make any changes.

To enable the non-repeating row feature, use the following procedure.

> **Note:** To use this feature, users must edit the HTML source directly. Microsoft Word does not allow direct HTML editing, so users will need to use a plain-text editor instead of Word. Also, Oracle has made every effort to keep Fulfillment HTML 4.01-compliant so that no user agents (browsers) will be affected by any of these changes.

1. Add new styles.

   Two empty styles identify rows as repeating (jtorep) and non-repeating (jtonorep). To specify that rows can be repeating or non-repeating, add the following styles between the <head> </head> tags of your HTML source document.

   ```
   <style type="text/css">
   jtonorep {}
   ```

```
jtorep {}
</style>
```

2. Optionally, change the default behavior.

   The default behavior is to expand automatically every row that contains merge fields. If your document has many rows with merge fields, but you want to repeat only some of those rows, you can change the default behavior to repeat only rows that contain merge fields when a tag specifies to repeat those rows.

   To change the default behavior, add the following <meta> tag to the <head> section of the HTML source document:

   ```
   <meta name="jto_tables" content="no_repeat">
   ```

   If this meta tag is not in the HTML source document, then the default behavior is used. Adding this tag changes only the default behavior.

3. Identify the repeating and non-repeating rows.

   After you have designed your document and you can see whether or not you want to set the default behavior to repeat, you are ready to modify the HTML table rows to indicate which rows should repeat and which rows should not repeat, as in the following example.

   If in the previous order form example, you want to add a single row at the bottom of the table to show the order total, you would have the following HTML code:

   ```
   <table border="1" summary="Order Table">
   <tr>
       <th>Item Number</th>
       <th>Item Description</th>
       <th>Quantity</th>
       <th>Unit Price</th>
       <th>Total Price</th>
   </tr>
   <tr>
       <td>{(ITEM_NUMBER)}</td>
       <td>{(ITEM_DESCRIPTION)}</td>
       <td>{(QUANTITY)}</td>
       <td>{(UNIT_PRICE)}</td>
       <td>{(TOTAL_PRICE)}</td>
       </tr>
       <tr>
       <td colspan="4">&nbsp;</td>
       <td><b>{(ORDER_TOTAL)}</b></td>
   ```

```
</tr>
</table>
```

The SQL query would change to:

```
SELECT   ITEM_NUMBER,
        ITEM_DESCRIPTION,
        QUANTITY,
        UNIT_PRICE,
        TOTAL_PRICE,
        ORDER_TOTAL
FROM ORDERS
WHEREORDER_NUMBER = :my_order_number
```

The default behavior repeats not only the item rows, but the order total row is repeated for each row that the query returns. To change the default, first add the new styles as described Step 1. Next, depending on whether or not you changed the default behavior described as in Step 2, you need to make one of the following changes.

Changing Default Behavior (Without Step 2 Addition)

The first row (the row with the table headers or <th> tags) does not have any merge fields, so it will not be repeated.

The second row will automatically be repeated because it contains merge fields, and you have not changed the default behavior. It can stay as it is.

The third row will also be repeated automatically, so you need to explicitly instruct it not to repeat. To do this, you need to use the jtonorep class style that was defined in step one. Change the third row from,

```
 <tr>
     <td colspan="4">&nbsp;</td>
     <td><b>{(ORDER_TOTAL)}</b></td>
 </tr>
```

to

```
<tr class="jtonorep">
    <td colspan="4">&nbsp;</td>
    <td><b>{(ORDER_TOTAL)}</b></td>
</tr>
```

This change specifies the class for this <tr> as `jtonorep`. When Fulfillment sees this tag, it will not repeat this row. It will only write the row once regardless of the number of rows that the query returns.

Altered Default Behavior Change

If you changed the default behavior as in Step 2, no rows will repeat. You need to tell Fulfillment which row to repeat.

If the second row is the row that you want to repeat, change it from,

```
<tr>
    <td>{(ITEM_NUMBER)}</td>
    <td>{(ITEM_DESCRIPTION)}</td>
    <td>{(QUANTITY)}</td>
    <td>{(UNIT_PRICE)}</td>
    <td>{(TOTAL_PRICE)}</td>
</tr>
```

to

```
<tr class="jtorep">
    <td>{(ITEM_NUMBER)}</td>
    <td>{(ITEM_DESCRIPTION)}</td>
    <td>{(QUANTITY)}</td>
    <td>{(UNIT_PRICE)}</td>
    <td>{(TOTAL_PRICE)}</td>
</tr>
```

The line `<class="jtorep">` was added to the <tr> element for the second row to tell Fulfillment to repeat this row for each result row from the SQL query.

Additional Information

Whether or not you changed the default behavior by including the <meta> tag from Step 2, you can use both attributes `class="jtonorep"` and `class="jtorep"` in the same document. These attributes override the default behavior and specify what to disregard. Also, these attributes only take affect where there are table rows with merge fields. If there are no merge fields, no rows are repeated.

Because of HTML 4.01 specifications, multiple class names can be associated to the class attribute. They simply need to be separated by white space. This is important if you have a nicely formatted HTML document already that is using style sheets. You only need to add `jtorep` or `jtonorep` to the necessary rows separated by white space.

Note: The class attribute must be in lowercase, and the class names must be enclosed in double quotes, as in: `<td class="jtorep">`

Conclusion

By using some standard HTML tags, Fulfillment can support both repeatable merge field rows, as originally designed, and non-repeatable rows that allows for more flexible formatting of Fulfillment documents using HTML tables.

HTML Examples

The following examples of HTML code demonstrate table order using the default behavior and the altered behavior.

Order Table Using the Default Fulfillment Behavior

```
<html>
<head>
    <title>Sample Order Table</title>
    <style type="text/css">
    jtonorep {}
    jtorep {}
    </style>
    </head>
    <body>
    <table border="1" summary="Order Table">
    <tr>
    <th>Item Number</th>
    <th>Item Description</th>
    <th>Quantity</th>
    <th>Unit Price</th>
    <th>Total Price</th>
    </tr>
    <tr>
    <td>{(ITEM_NUMBER)}</td>
    <td>{(ITEM_DESCRIPTION)}</td>
    <td>{(QUANTITY)}</td>
    <td>{(UNIT_PRICE)}</td>
    <td>{(TOTAL_PRICE)}</td>
    </tr>
    <tr class="jtonorep">
    <td colspan="4">&nbsp;</td>
    <td><b>{(ORDER_TOTAL)}</b></td>
    </tr>
    </table>
</body>
</html>
```

Order Table Example With Altered Behavior

```
<html>
<head>
    <title>Sample Order Table</title>
    <meta name="jto_tables" content="no_repeat">
    <style type="text/css">
    jtonorep {}
    jtorep {}
    </style>
</head>
<body>
    <table border="1" summary="Order Table">
    <tr>
    <th>Item Number</th>
    <th>Item Description</th>
    <th>Quantity</th>
    <th>Unit Price</th>
    <th>Total Price</th>
    </tr>
    <tr class="jtorep">
    <td>{(ITEM_NUMBER)}</td>
    <td>{(ITEM_DESCRIPTION)}</td>
    <td>{(QUANTITY)}</td>
    <td>{(UNIT_PRICE)}</td>
    <td>{(TOTAL_PRICE)}</td>
    </tr>
    <tr>
    <td colspan="4">&nbsp;</td>
    <td><b>{(ORDER_TOTAL)}</b></td>
    </tr>
    </table>
</body>
</html>
```

See Also

HTML 4.01 Specification http://www.w3.org/TR/html4/

7.4.16 Server id and User id in create_fulfillment API Call

Is it necessary to pass Server id and User id for the create_fulfillment API call?

Server_id is not required, but user id is required.

In Release 11.5.8, if a valid One-to-One Fulfillment Server was not associated with the user, API would throw an error. In Release 11.5.9, API checks for the One-to-One Fulfillment Server that is associated with that user. If the API does not find any server associated with the user, then the request is submitted to Server Id 5000 (the Rapid Install One-to-One Fulfillment Server that is available out of the box). The Rapid Install Server must be running to pick up these requests.

7.4.17 Sending a Request Through Multiple Media Types

How can I send the same request through multiple types of media, such as e-mail, fax, and print?

Fulfillment allows sending a document by using the supported media types EMAIL, FAX and PRINT.

When GET_CONTENT_XML is called, the p_media_type parameter can be set to the following combinations of values, depending on the requirements.

- p_media_type := 'EMAIL' or 'FAX' or 'PRINTER'
- or 'EMAIL,FAX,PRINTER'
- or 'EMAIL, FAX'
- or 'EMAIL, PRINTER'
- or 'FAX,PRINTER'

The sequence does not matter. Separate the media types with a comma. To successfully send a message, ensure that proper values are passed for the parameters p_emai, p_fax and p_printer.

7.5 API Error Messages

The following table lists some common API error messages and their causes.

Table 7–3 Common API Error Messages

Package	Procedure	Notes	Error Codes	Error Text
JTF_FM_OCM_ REND_REQ				
	CREATE_ FULFILLMENT_ RENDITION			
		error messages specific to the request of type single		
			JTF_FM_API_ MISSING_PRINT_ ADDR	Chosen Media of Output is Print but missing print address
			JTF_FM_API_ MISSING_FAX_ADDR	Chosen Output Media is Fax but missing fax address
			JTF_FM_API_ MISSING_EMAIL_ ADDR	Chosen Media of output is Email but missing Email address
		error messages specific to 'Test' requests		
			JTF_FM_API_TEST_ MISS_PRINT_ADD	"This ""Test"" print request is missing printer address. Please pass valid printer address to successfully submit the ""Test"" request."
			JTF_FM_API_TEST_ MISS_EMAIL_ADD	"This ""Test"" email request is missing email address. Please pass valid email address to successfully submit the ""Test"" request."

Table 7–3 Common API Error Messages (Cont.)

Package	Procedure	Notes	Error Codes	Error Text
			JTF_FM_API_TEST_ MISS_FAX_ADD	"This ""Test"" fax request is missing fax address. Please pass valid fax address to successfully submit the ""Test"" request."
		common to all requests submitted through create_ fulfillment_ Rendition		
			JTF_FM_API_CF_ INVALID_MEDIA	"Invalid Media Type specified. Only allowed Media types are any one of the following. ""EPF,EFP,FEP,FPE,PEF,PFE,EP ,EF,E,PE,PF,P,FE,FP,F"". E represents ""EMAIL"", P represents ""PRINTER"", and F represents ""FAX"""
			JTF_FM_API_ MISSING_BATCH_LIST	The Value of the Batch List can not be null.
			JTF_FM_OCM_ CONTENT_VER_ABS	Could not find Renditions for content with citem_id = &ARG1 and citem_ver_id = &ARG2
			JTF_FM_API_TXT_ HTML_ABS	Email/Fax requests that are sent through fulfillment need either a HTML rendition or TEXT rendition. The cover letter with citem_id = &ARG1 and citem_ver_id = &ARG2 does not have HTML or TEXT rendition. Please use a cover letter with a valid rendition
			JTF_FM_API_ MISSING_CONTENT_ ID	Value for Content Id cannot be NULL

Table 7–3 Common API Error Messages (Cont.)

Package	Procedure	Notes	Error Codes	Error Text
			JTF_FM_API_ INVALID_MEDIATYPE	"Invalid media type &ARG1 passed for Content Id &ARG2. Allowed media types are EMAIL, FAX, PRINTER and PATH."
			JTF_FM_API_ MISSING_MEDIA_ TYPE	Value for Media Type cannot be NULL
			JTF_FM_OCM_ NOTAPP_OR_ABS	This OCM content &ARG1 is either Absent or is not approved
			JTF_FM_GET_TRANS_ ITEM_FAIL	Call to IBC_CITEM_ADMIN_ GRP.get_trans_item API failed for CITEM_VER_ID : &ARG1
			JTF_FM_API_ INVALID_REQTYPE	"Invalid request type specified. Only allowed values are ""E,P,S or T"". E for ""Electronic Request"" P for ""Physical Collateral"" , T for ""Test Requests"" and S for Single Requests"
			JTF_FM_API_ MISSING_OCM_ QUERY	Template &ARG1 must have a valid query associated with it to be fulfilled.
JTF_FM_ REQUEST_GRP				
	GET_CONTENT_ XML	"consumers who call Fulfillment through the triad. (Call Start Request, Get_ Content_Xml and then submit_ request)		"

Table 7–3 Common API Error Messages (Cont.)

Package	Procedure	Notes	Error Codes	Error Text
			JTF_FM_API_ MISSING_CONTENT_ ID	Value for Content Id cannot be NULL
			JTF_FM_API_ MISSING_MEDIA_ TYPE	Value for Media Type cannot be NULL
			JTF_FM_API_ MISSING_ CONTENTTYPE	Value for Content Type cannot be NULL
			JTF_FM_API_ MISSING_REQUEST_ ID	Value for Request Id cannot be NULL
			JTF_FM_API_ INVALID_MEDIATYPE	"Invalid media type &ARG1 passed for Content Id &ARG2. Allowed media types are EMAIL, FAX, PRINTER"
			JTF_FM_OCM_ NOTAPP_OR_ABS	This OCM content &ARG1 is either Absent or is not approved
			JTF_FM_EXCEPTION_ IN_GET_ITEM	An Unexpected exception was caught when IBCs GET_ITEM API was called for content item &ARG1
			JTF_FM_GET_TRANS_ ITEM_FAIL	Call to IBC_CITEM_ADMIN_ GRP.get_trans_item API failed for CITEM_VER_ID : &ARG1
	CANCEL_ REQUEST			
			JTF_FM_API_ ALREADY_ CANCELED	"Request Id &ARG1 is already in ""CANCELLED"" state."
			JTF_FM_API_ MISSING_REQUEST_ ID	Value for Request Id cannot be NULL
			JTF_FM_API_ REQUEST_ NOTFOUND	Could not find fulfillment request for Request Id &ARG1 in the database

Table 7–3 Common API Error Messages (Cont.)

Package	Procedure	Notes	Error Codes	Error Text
			JTF_FM_API_QUEUE_ NOTFOUND	Fulfillment API Msg: Please check the setup for fulfillment server. Contact your system administrator. Specific Problem - Could not find queue names for Server Id &ARG1
	SUBMIT_BATCH_ REQUEST			
			JTF_FM_API_ MISSING_LIST_TYPE	Value for List Type cannot be NULL
			JTF_FM_API_ MISSING_CONTENT_ XML	Value for Content XML cannot be NULL
			JTF_FM_API_ MISSING_VIEW_ NAME	Value for View Name cannot be NULL
			JTF_FM_API_ MISSING_BATCH_LIST	The Value of the Batch List can not be null.
			JTF_FM_API_ INVALID_LISTTYPE	Invalid list type &ARG1 passed. Allowed list types are VIEW or ADDRESS
	SUBMIT_MASS_ REQUEST			
			JTF_FM_API_ MISSING_CONTENT_ XML	Value for Content XML cannot be NULL
			JTF_FM_API_ MISSING_OCM_ QUERY	Template &ARG1 must have a valid query associated with it to be fulfilled
		common to all requests within triad.		
			JTF_FM_API_ MISSING_USER_ID	Value for User Id cannot be NULL

Table 7–3 Common API Error Messages (Cont.)

Package	Procedure	Notes	Error Codes	Error Text
			JTF_FM_API_ MISSING_CONTENT_ XML	Value for Content XML cannot be NULL
			JTF_FM_API_ MISSING_REQUEST_ ID	Value for Request Id cannot be NULL
			JTF_FM_API_ REQUESTID_REUSED	Cannot use a previously submitted request ID to submit a new request
			JTF_FM_API_ MISSING_SERVER_ID	Value for Server Id cannot be NULL
			JTF_FM_API_QUEUE_ NOTFOUND	Fulfillment API Msg: Please check the setup for fulfillment server. Contact your system administrator. Specific Problem - Could not find queue names for Server Id &ARG1
	RESUBMIT_ REQUEST		JTF_FM_API_ MISSING_REQUEST_ ID	Value for Request Id cannot be NULL
			JTF_FM_API_ REQUEST_ NOTFOUND	Could not find fulfillment request for Request Id &ARG1 in the database
			JTF_FM_API_QUEUE_ NOTFOUND	Fulfillment API Msg: Please check the setup for fulfillment server. Contact your system administrator. Specific Problem - Could not find queue names for Server Id &ARG1
	RESUBMIT_JOB			
			JTF_FM_API_ MISSING_JOB_ID	Job Id &ARG1 not found for request Id &ARG2
			JTF_FM_API_ MISSING_MEDIA_ TYPE	Value for Media Type cannot be NULL

Table 7–3 Common API Error Messages (Cont.)

Package	Procedure	Notes	Error Codes	Error Text
			JTF_FM_API_INV_ MEDIA_TYPE	"The media type passed is invalid. Valid media types are ""EMAIL"", ""FAX"" and ""PRINTER"""
			JTF_FM_API_ MISSING_REQUEST_ ID	Value for Request Id cannot be NULL
			JTF_FM_API_ REQUEST_ NOTFOUND	Could not find fulfillment request for Request Id &ARG1 in the database
	CORRECT_ MALFORM_JOB			
			JTF_FM_API_ MISSING_REQ_INFO	The number of JOB IDs must be the same as the list of addresses
	RESUBMIT_ MALFORMED			
			JTF_FM_NO_MAL_ CORRECTION_ FOUND	Could not find any corrected address for the malformed request

7.6 Debugging

Use the following procedures to set and view debug logging for PL/SQL and the Fulfillment Server.

7.6.1 PL/SQL Debugging

All logging is at the procedure level (Level 2). To debug at the API level, turn on debugging, and set the logging level to Procedure or Statement.

To run a test script and enable debugging, call the following API before calling the Fulfillment APIs.

```
l_session_id          NUMBER :=1000;
fnd_global.initialize(l_session_
id,1000,1000,1000,1000,1000,1000,100,100,100,100,1000);
```

After the logging level is set and the application is initialized, whenever a request is sent to the One-to-One Fulfillment API, a debug is written to the central repository.

To view all debugs, use the following procedure.

Login

HTML Administration

Responsibility

System Administrator

Prerequisites

None

Steps

1. Select the Settings tab > System sub tab > **Messages** link in the Debug Logging group.

   The Debug Log Viewer page appears.

2. In the fields, enter search criteria to define the search range.

3. In the Module field, reset the debug logging level by entering **`jtf.plsql%JTF_FM%`** .

4. Click **View Log**.

   The Log Summary and Log Messages appear.

7.6.2 Server Debugging

For the server to write to the central logging repository, set the logging level to one of the following levels:

- Event (Level 3)
- Procedure (Level 2)
- Statement (Level 1)

After a request is processed, use the following procedure to view logs.

Login

HTML Administration

Responsibility

System Administrator

Prerequisites

None

Steps

1. Select the Settings tab > System sub tab > **Messages** link in the Debug Logging group.

   The Debug Log Viewer page appears.

2. In the fields, enter search criteria to define the search range.

3. Do one of the following:

   - In the Module field, reset the debug logging level by entering **`jtf.fm.engine.logging%`**.
   - If the request identifier is known, append it to the Module value, for example, `jtf.fm.engine.logging%12345%` (assuming 12345 is the request identifier).
   - If the job is also known and the request identifier, add it to the Module value also, for example, `jtf.fm.engine.logging%12345%34%` (assuming 35 is the party identifier).

7.7 Troubleshooting the RightFax Integration

Q: What can I do if a fax request is sent and the Fulfillment Server is not set up to send faxes?

A: If the Fulfillment Server receives a fax request and is not configured to send faxes, an error will be logged in the Servers error log and Fulfillment History tables will note the failure to send the fax.

Q: What can I do if a print request is sent and the Fulfillment Server is not set up to send print jobs?

A: If the Fulfillment Server receives a print request and is not configured to send print jobs, an error will be logged in the Servers error log and Fulfillment History tables will note the failure to print.

8 Administration Tasks

This chapter describes task-based procedures for administering Oracle One-to-One Fulfillment. Topics include:

- Managing Servers
- Defining Print Servers and Printers for the High-Quality Printing
- Managing Fulfillment Agent Groups
- Managing Oracle Marketing Encyclopedia System Templates
- Managing Oracle Marketing Encyclopedia System Master Documents
- Viewing Oracle Marketing Online Collateral
- Managing Oracle Marketing Encyclopedia System Queries
- Managing Content
- Managing Fulfillment Requests
- Managing Fulfillment Request History
- Using the Quick Find Feature
- Using Merge Fields
- Setting Up High Quality Printing

8.1 Managing Servers

Use the following procedures to administer the Fulfillment, e-mail, print and fax servers.

- Viewing Fulfillment Server Status

- Setting the Fulfillment Server Start Time and Shutdown Time
- Setting the Interaction History Outcomes and Results
- Enabling the Unsubscribe Option in E-Mail Requests
- Enabling Tracking for E-Mail Requests

> **Note:** In all Fulfillment server configuration options, for the changes to take effect you must restart any running servers. The information is gathered only once at startup.

8.1.1 Viewing Fulfillment Server Status

Use the following procedure to determine whether or not the Fulfillment Server is running.

Login

HTML Login URL

Responsibility

Call Center HTML Administration

Prerequisites

Define at least one Fulfillment Server

Steps

1. Select the ICSM tab > Server Groups sub tab.

   The Server Group List page appears.

2. Click the Server Group Name that contains the Fulfillment Server.

   The Server Group Details page appears.

3. In the Servers group, look in the Status column of the Fulfillment Server. If the flag is green, then the server is running. If the flag is red, then the server is down. Tool tips also display the server status.

8.1.2 Setting the Fulfillment Server Start Time and Shutdown Time

Use this procedure to set the start and shutdown times for the Fulfillment Server. The start and shutdown times indicate the times when the Fulfillment Server polls the fulfillment request queues in the database. It does not physically start or stop the server. You can set a start and shutdown time if you do not want the Fulfillment Server to process fulfillment requests twenty-four hours a day.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Server tab > Server sub tab.

   The Servers page appears.

2. Click a Fulfillment Server name link.

   The Server Details page appears.

3. From the Start Time lists, select the hour and minute for the time at which the Fulfillment Server starts polling the fulfillment request queues in the database. It does not physically start the server. If running, the server will continue to process retrieved fulfillment requests even when it is not polling the database.

4. From the Shutdown Time lists, select the hour and minute for the time at which the Fulfillment Server stops polling the fulfillment request queues in the database. It does not physically stop the server. If running, the server will continue to process retrieved fulfillment requests even when it is not polling the database.

5. Click **Continue**.

6. Restart any currently running Fulfillment servers for the changes to take effect.

8.1.3 Setting the Interaction History Outcomes and Results

Use this procedure to set the interaction history outcomes and results for processed fulfillment requests.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Server tab > Server sub tab.

   The Servers page appears.

2. Click a Fulfillment Server name link.

   The Server Details page appears.

3. From the Request Outcome list, select the outcome that you want to associate with a fulfillment request job that has been processed by the Fulfillment Server.
4. From the Request Success Result list, select the result that you want to associate with a fulfillment request job that has been successfully processed by the Fulfillment Server.
5. From the Request Failure Result list, select the result that you want to associate with a fulfillment request job that has not been successfully processed by the Fulfillment Server.
6. Click **Continue**.
7. Restart any currently running Fulfillment servers for the changes to take effect.

8.1.4 Enabling the Unsubscribe Option in E-Mail Requests

Use this procedure to enable the Fulfillment Server to insert a hypertext link into the e-mail content that will allow recipients to access their contact preferences in Oracle Marketing.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Server tab.

   The Servers page appears.

2. Click a Fulfillment Server Name.

   The Server Details page appears.

3. In the side panel, click **Email Servers**.

   The Email Servers page appears.

4. Select **Unsubscribe**.

5. In the Unsubscribe Text and URL field, enter the message that explains to customers how to unsubscribe. Enter the URL by which customers can unsubscribe.

6. Click **Update**.

7. Restart any currently running Fulfillment servers for the changes to take effect.

8.1.5 Enabling Tracking for E-Mail Requests

Use this procedure to enable the Fulfillment Server to track e-mail statistics for all e-mail requests.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Server tab.

   The Servers page appears.

2. Click a Fulfillment Server name link.

   The Server Details page appears.

3. In the side panel, click **Email Servers**.

   The E-mail Servers page appears.

4. Select **Track Email**.
5. In the Tracking URL field, enter the HTML code that includes the URL that is called when recipients open e-mail.
6. Click **Update**.
7. Restart any currently running Fulfillment servers for the changes to take effect.

8.1.6 Setting Up Bounce Back

Bounce back management allows users to address outbound e-mail messages so that destination e-mail servers send "bounces" back to a specific account. The address to which the bounced messages are sent contain the request and job information, which allows Oracle One-to-One Fulfillment to identify which e-mail messages bounced.

E-mail administrators need to update the Sendmail configuration to direct bounced messages into a single bounce back account. The configuration steps that are required for Bounce Back Management can be performed on both new and existing Fulfillment Servers. Bounce back management does not affect the functionality that is provided with Tracking or with the Unsubscribe feature.

Bounced messages are recorded. To determine the reason why a message bounced, a user would have to perform an analysis outside of what Oracle One-to-One Fulfillment provides.

8.1.6.1 Configuring Bounce Back

Use the following procedure to configure e-mail bounce back.

Login

Not Applicable

Responsibility

Not Applicable

Prerequisites

None

Steps

1. The e-mail administrator must create a bounce back account, and then record the host name of the IMAP server, the user name for the bounce back account, and the password. For example, assume the name of the account is "bounces."

   Modify sendmail to reroute messages addressed to the name of the bounce account plus any addition string to the bounce account. For example, if your bounce account is called bounces@domain.com, make sure that any e-mail messages addressed to bounces_12345@domain.com get routed to the bounce account. An example sendmail rule to do this is shown below, but this task is left to the mail administrator.

   ```
   Rbounces _$* < @ $* > bounces< @ $ 2 >
   ```

2. Restart Sendmail.

3. Test by sending a message to bounces_123@<bouncedomain>.com and check that it arrives in the bounce account.

   The message should arrive in the bounce account. The configuration and testing of the Sendmail setup is up to the local e-mail administrator.

   When Sendmail is redirecting bounced e-mail messages to the bounce account, you can set up the Fulfillment Server.

8.1.6.2 Configuring Bounce Back In Fulfillment Administration Console

Use the following procedure to configure e-mail bounce back in the Administration console.

Login

Fulfillment Administrator Console

Responsibility

Not Applicable

Prerequisites

Set up Sendmail

Steps

1. Select the Server tab > server name.

   The Email Servers page appears.

2. In the Bounceback Management group, select the Enable Bounceback option.

   This option instructs the Fulfillment Server to create the bounceback monitoring thread when it starts, and to append the bounceback information as the From address on outgoing e-mail messages.

3. Enter the host name or IP address of the IMAP host where the bounce account has been created.
4. Add the IMAP port. The default IMAP port is 143.
5. Enter the account name and the account password.
6. Select a run frequency. The run frequency indicates how often the bounceback monitoring thread will run. A typical Run Frequency is every two hours. Set this value lower if you need quicker results and are sending large volumes of e-mail. Set this value higher if you are not concerned with recording bounceback messages. The difference is that they will be recorded, but if you only have it run every twelve hours, updates to the counts are only done in that interval.

   When running, the thread connects to the bounces account and scans the inbox for any new messages. For each new message, it updates the database. When you view the request history, you can see the number of bounced messages in a particular request.

7. Enter a Return Path. The return path should be the e-mail address for the bounce account, for example, bounces@yourcompany.com. Oracle One-to-One Fulfillment uses all information to the left of the @ sign as the prefix that you should use in your sendmail.cf file.
8. Restart the Fulfillment server for the changes to take effect.

8.2 Defining Print Servers and Printers for the High-Quality Printing

The Printer tab is only for users of Oracle E-Business Suite Applications to configure print servers and printers that are used with the High-Quality Printing.

Tasks

You can perform the following tasks:

- Configuring a Print Server for the High-Quality Printing
- Configuring Printers for the High-Quality Printing

8.2.1 Configuring a Print Server for the High-Quality Printing

Use the following procedure to create a print server.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Select the Printer tab.

   The Print Servers page appears.

2. Click **Create**.

   The Create Print Server page appears.

3. In the Print Server Name field, type a descriptive name for the print server.
4. For printing RTF or PDF documents, in the Connection String field, type the URL or IP address of the print server.
5. From the Server Type list, select **Pasta**.
6. Click **Apply**.

   The Print Servers page appears.

See Also

To associate a printer server with the Fulfillment Server, see the *Oracle One-to-One Fulfillment Implementation Guide*.

8.2.2 Configuring Printers for the High-Quality Printing

Use the following procedure to associate printers with a printer server.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Select the Printer tab.

   The Print Servers page appears.

2. Click the print server Name.

   The Update Print Server page appears.

3. Click **Create Printer**.

   The Create Printer page appears.

4. In the Printer Name field, type a descriptive name for this printer.
5. Optionally, in the Description field, type a description of this printer.
6. In the Network Name field, type the printer name that identifies this printer on your local area network.
7. Click **Apply**.

   The Update Print Server page appears.

See Also

To associate a printer server with the Fulfillment Server, see the *Oracle One-to-One Fulfillment Implementation Guide*.

8.3 Managing Fulfillment Agent Groups

Use the Group sub tab to create a fulfillment agent group and assign the fulfillment agent group to a Fulfillment Server.

Tasks

You can perform the following tasks:

- Creating an Agent Group
- Adding an Agent to a Group
- Removing an Agent from a Group
- Assigning a Fulfillment Agent Group to a Fulfillment Server

8.3.1 Creating an Agent Group

Use this procedure to define a group of agents for fulfillment processing.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Create user accounts for agents.

Steps

1. Click the Group tab > Groups sub tab.

   The Groups page appears.

2. Click **Create**.

   The Create Group page appears.

3. In the Group Name field, enter a unique name for the group.

   You cannot change this field after the server is saved.

4. Optionally, in the Description field, enter a description of the group.

   This field is for informational purposes only.

5. From the Server Name list, select the name of the Fulfillment Server.

> **Note:** Each fulfillment agent group must be linked to a Fulfillment Server. However, if you have not created a Fulfillment Server, you can return to the Group tab and select a Fulfillment Server later.

6. Add agents to the group.

   a. In the Agent area, in the Name column, type a user name or other search criteria, and then click **Go**.

   You may search for a user name using character strings and the wildcard symbol (%). For example, to find user names starting with A, type A%.

   The Select Agent page appears. The search results are listed.

   b. To modify your search, type new search criteria in the Search field and then click **Search**.

   c. Click a user name.

   The Create Group page appears. The user name appears in the Agents area.

7. Click **Create**.

   The Groups page appears.

8.3.2 Adding an Agent to a Group

Use this procedure to add an agent to a fulfillment agent group.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Group tab > Groups sub tab.

The Groups page appears.

2. Click a group name.

   The Group Detail page appears.

3. In the Agent area, in the Name column, type a user name or other search criteria, and then click **Go**.

   You may search for a user name using character strings and the wildcard symbol (%). For example, to find user names starting with A, type A%.

   To clear the search criteria, click the Restore button.

   The Select Agent page appears. The search results are listed.

4. To modify the search, type new search criteria in the Search field and then click **Search**.

5. Click a user name.

   The Group Detail page appears. The user name appears in the Agents area.

6. Click **Update**.

8.3.3 Removing an Agent from a Group

Use this procedure to remove an agent from a Fulfillment agent group.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Group tab.

2. Click the Groups sub tab.

   The Groups page appears.

3. Click a group name.

The Group Detail page appears.

4. In the Agent area, select the Remove box for the agent that you want to remove from the agent group.
5. Click **Update**.

   The agent is removed from the agent group and the Groups page appears.

8.3.4 Assigning a Fulfillment Agent Group to a Fulfillment Server

Use this procedure to assign a fulfilment agent group to a Fulfillment Server.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

- Create user accounts for agents.
- Create a fulfillment agent group.
- Configure a fulfillment server.

Steps

1. Click the Group tab > Groups sub tab.

   The Groups page appears.
2. Click a group name.

   The Group Detail page appears.
3. From the Server Name list, select a Fulfillment Server.
4. Click **Update**.

8.4 Managing Oracle Marketing Encyclopedia System Templates

Use the Template sub tab to manage groupings of Oracle Marketing Encyclopedia System master documents and OMO collateral items.

Tasks

You can perform the following tasks:

- Creating a Template
- Assigning a Master Document to a Template
- Assigning Collateral to a Template
- Updating a Template
- Removing a Master Document from a Template
- Removing Collateral from a Template

8.4.1 Creating a Template

Use this procedure to create a grouping of Oracle Marketing Encyclopedia System master documents and Oracle Marketing collateral items (also called deliverables).

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Create master document and at least one master document collateral.

Steps

1. Click the Template tab > Template sub tab.

   The Templates page appears.

2. Click **Create**.

   The Create Template page appears.

3. Enter the details about the template.

   a. In the Template Name field, type a unique name for the template.

   b. From the Status list, choose the status of the template.

      You have the following options:

- * Active: The template is available for fulfillment requests.
- * Inactive: The template is not available for fulfillment requests.

c. Optionally, in the Description, type a description of the template.

This field is for informational purposes only.

d. Click **Continue**.

The Template Master Document window opens.

4. Add one or more master documents to the template.

a. In the Master Documents area, in the Name column, type a master document name or other search criteria, and then click **Go**.

You may search for a master document using character strings and the wildcard symbol (%). For example, to find master documents starting with A, A%.

To clear the search criteria, click **Restore**.

The Select Master Document page appears. The search results are listed.

b. To modify your search, type new search criteria in the Search field and then click **Search**.

c. Click **Select** next to the master document name.

d. Click **Update**.

The View Template page appears.

5. Optionally, to add one or more collateral to the template, click **Collateral**.

a. In the Collateral area, in the Name column, type a collateral name or other search criteria, and then click **Go**.

You may search for collateral using character strings and the wildcard symbol (%). For example, to find collateral starting with A, A%.

If you want to clear the search criteria, then click **Clear**.

The Select Collateral page appears, listing the search results.

b. To modify your search, type new search criteria in the Search field and then click **Search**.

c. Click **Select** next to the collateral name.

d. Click **Update**.

The View Template page appears.

8.4.2 Assigning a Master Document to a Template

Use this procedure to associate a master document with a template.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

- Create a template.
- Create a master document.

Steps

1. Click the Template tab > Template sub tab.

   The Templates page appears.

2. Click a template name.
3. In the side panel, click Master Documents.

   The View Template page appears.

4. In the Master Documents area, in the Name column, type a master document name or other search criteria, and then click **Go**.

   You may search for a master document using character strings and the wildcard symbol (%). For example, to find master documents starting with A, A%.

   To clear the search criteria, click **Restore**.

   The Select Master Document page appears. The search results are listed.

5. If you want to modify your search, type new search criteria in the Search field and then click **Search**.
6. Click **Select** next to a master document name.
7. Click **Update**.

   The View Template page appears.

8.4.3 Assigning Collateral to a Template

Use this procedure to associate a collateral item with a template.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

- Create a template.
- Create collateral.

Responsibility

One-to-One Fulfillment Administrator

Steps

1. Click the Template tab > Template sub tab.

   The Templates page appears.

2. Click a template name.
3. In the side panel, click Collateral.

   The View Template page appears.

4. In the Collateral area, in the Name column, type a collateral name or other search criteria, and then click **Go**.

   You may search for collateral using character strings and the wildcard symbol (%). For example, to find collateral starting with A, A%.

   To clear the search criteria, click **Restore**.

   The Select Collateral page appears. The search results are listed.

5. To modify your search, type new search criteria in the Search field and then click **Search**.
6. Click **Select** next to a collateral name.
7. Click **Update**.

The View Template page appears.

8.4.4 Updating a Template

Use this procedure to update a template.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Template sub tab.

   The Templates page appears.

2. Click a template name link.

   The View Template page appears.

3. From the Status list, select the status of the template.

   You have the following options:

   - Active: The template is available for fulfillment requests.
   - Inactive: The template is not available for fulfillment requests.

4. Optionally, in the Template Description field, type a description about the template.

   This field is for informational purposes only.

5. Click **Update**.

8.4.5 Removing a Master Document from a Template

Use this procedure to remove a master document from a template.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Template sub tab.

   The Templates page appears.

2. Click a template name.
3. In the side panel, click **Master Documents**.

   The View Template page appears.

4. Select the Remove box for the master document that you want to remove from the template.
5. Click **Update**.

   The master document is disassociated from the template and the Master Document list in the View Template page is refreshed.

8.4.6 Removing Collateral from a Template

Use this procedure to disassociate a collateral item from a template.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Template sub tab.

   The Templates page appears.

2. Click a template name.
3. In the side panel, click **Collateral**.

   The View Template page appears.

4. Select **Remove** next to the collateral name.
5. Click **Update**.

   The collateral is disassociated from the template. The Collateral list in the View Template page is refreshed.

8.5 Managing Oracle Marketing Encyclopedia System Master Documents

Use the Master Document sub tab to manage documents that are stored in Oracle Marketing Encyclopedia System. The content in the document becomes the body of the fulfillment e-mail or fax print job.

Tasks

You can perform the following tasks:

- Creating an Oracle Marketing Encyclopedia System Master Document Using the Text Editor
- Uploading an Oracle Marketing Encyclopedia System Master Document
- Updating an Oracle Marketing Encyclopedia System Master Document
- Viewing an Oracle Marketing Encyclopedia System Master Document
- Downloading an Oracle Marketing Encyclopedia System Master Document
- Assigning a Query to an Oracle Marketing Encyclopedia System Master Document

- Removing a Query From an Oracle Marketing Encyclopedia System Master Document

8.5.1 Creating an Oracle Marketing Encyclopedia System Master Document Using the Text Editor

Use this procedure to create a master document using the text editor in Oracle One-to-One Fulfillment. The master document will be saved in Oracle Marketing Encyclopedia System.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Master Document sub tab.

   The Master Document page appears.

2. Click **Create and Upload**.

   The Master Document Editor page appears.

3. In the Master Document Name field, type a unique name for the master document.

   When you save the document, the body of the fulfillment document will be saved with this name (for example, `name.htm`). In addition, the name will appear in list of documents in Oracle One-to-One Fulfillment and the business application. You cannot change this field after the document is saved.

4. If you want to compose the master document using plain text, click the **Change to Plain Text Mode** link.

5. If you want to compose the master document using formatted text, click the **Change to Rich Text Mode** link.

   A formatting tool bar appears.

6. If you want to insert a merge field derived from a query, then do the following:

   a. In the SQL Merge Field field, click **Create Query**.

      The SQL Editor page appears.

   b. In the Query Name field, type a unique name for the query.

      You cannot change this field after the server is saved.

   c. Optionally, in the Query Description field, enter a description of the query.

      This field is for informational purposes only.

   d. In the SQL Query field, type the SQL statement.

      Note: Do not insert a semicolon (;) at the end of the SQL statement.

   e. Click **Validate SQL and Save**.

      If the syntax is valid, the SQL Editor page closes and the Master Document Editor page refreshes. The text of the query appears in the SQL Query field. The aliases (or columns, if there are no aliases) in the query appear in the SQL merge field list. The query is saved in Oracle Marketing Encyclopedia System and appears in the Query list.

      If you want to edit the SQL statement, click the **Edit** link.

      Merge fields are the fields available from the query.

   f. From the SQL/Merge Field list, select a merge field.

   g. Place the cursor in the document area.

   h. Click **Insert**.

7. If you want to manually create a merge field, then in the Custom Merge Field, type a merge field from the query and then click **Insert**.

   The merge field is inserted into the document.

8. Click **Save**.

   The Master Documents page appears. The master document appears in the list and if a query was created when the master document was created, the query appears in the Query list. The document is saved with an htm extension.

8.5.2 Uploading an Oracle Marketing Encyclopedia System Master Document

Use this procedure to upload a document to the Oracle Marketing Encyclopedia System as a master document.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Master Document sub tab.

   The Master Document page appears.

2. Click **Upload**.

   The Update Master Document page appears.

3. In the File Name field, click the **Browse** button to navigate to the directory that contains the document.

   Note that the name of the file is the name of the document that will appear in the Master Documents list. You will not be able to change this name later.

4. Select the master document.
5. Click **Upload**.

   The Upload Master Document page refreshes.

6. Optionally, in the Description field, type a description of the Master Document.
7. Optionally, from the Query list, select the query that you want to associate with the master document.
8. Click **Upload**.

   The Master Document page appears.

8.5.3 Updating an Oracle Marketing Encyclopedia System Master Document

To update a document that has been uploaded to Oracle Marketing Encyclopedia System, upload the document again. The document will have a unique numerical identifier even if it has the same name as a previously uploaded document. Therefore, the list of Oracle Marketing Encyclopedia System master documents may contain more than one document with the same name.

8.5.4 Viewing an Oracle Marketing Encyclopedia System Master Document

Use this procedure to view an Oracle Marketing Encyclopedia System master document.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Master Document sub tab.

   The Master Document page appears.

2. Click a master document name link.
3. If the document is an HTML file, then the file appears in a new browser window.
4. If the document is a Zip file, then a File Download window appears.

   You can open or save the Zip file.

8.5.5 Downloading an Oracle Marketing Encyclopedia System Master Document

Use this procedure to download an Oracle Marketing Encyclopedia System master document to a local file system.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Master Document sub tab.

   The Master Document page appears.

2. Right-click a master document name link.

   You can open or save the file.

8.5.6 Assigning a Query to an Oracle Marketing Encyclopedia System Master Document

Use this procedure to assign a query to an Oracle Marketing Encyclopedia System master document.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Master Document sub tab.

   The Master Document page appears.

2. From the Query list, select a query.

8.5.7 Removing a Query From an Oracle Marketing Encyclopedia System Master Document

Use this procedure to remove a query from an Oracle Marketing Encyclopedia System master document.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Template tab > Master Document sub tab.

   The Master Document page appears.

2. From the Query list, select **No Query**.

8.6 Viewing Oracle Marketing Online Collateral

Use the Template tab > Collateral sub tab to view a list of the collateral items (called "deliverables" in Oracle Marketing) that are available in Oracle Marketing Online. Collateral items are attached to Fulfillment e-mail.

8.7 Managing Oracle Marketing Encyclopedia System Queries

Use the Query sub tab to manage queries for Oracle Marketing Encyclopedia System Master Documents. These queries supply data for dynamic Oracle Marketing Encyclopedia System master documents.

Tasks

You can perform the following tasks:

- Creating a Query
- Updating a Query

- Viewing Oracle Marketing Encyclopedia System Master Documents Associated With an Query
- Creating a Data Source File from a Query

8.7.1 Creating a Query

Use this procedure to create a query.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Query tab > Query sub tab.

   The Queries page appears.

2. Click **Create**.

   The Create Query page appears.

3. In the Query Name field, type a unique name for the query.

   You cannot change this field after the query is saved.

4. Optionally, in the Description field, enter a description of the query.

   This field is for informational purposes only.

5. In the SQL Query field, type the SQL SELECT statement.

6. Click **Validate SQL**.

   If the syntax of the statement is valid and there are no bind variables, then the query is saved and Queries page appears.

   If there are bind variables, then you will have to supply the bind variable before the query can be saved.

8.7.2 Updating a Query

Use this procedure to update a query.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Query tab > Query sub tab.

   The Queries page appears.

2. Click a query name link.

   The Create Query page appears.

3. Optionally, in the Description field, enter a description of the query.

   This field is for informational purposes only.

4. In the SQL Query field, type the SQL SELECT statement.
5. Click **Validate SQL**.

8.7.3 Viewing Oracle Marketing Encyclopedia System Master Documents Associated With an Query

Use this procedure to view a list of the Oracle Marketing Encyclopedia System Master Documents that use a query.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Query tab > Query sub tab.

   The Queries page appears.

2. Click a query name link.

   The Create Query page appears. The master documents that use the query appear under the Documents Using Query heading.

8.7.4 Creating a Data Source File from a Query

Use the data source sub tab to download a list of merge fields for a query. The text file can be used by other tools, such as Microsoft Word, to create a master document that can be uploaded to Oracle Marketing Encyclopedia System.

Use this procedure to download a list of merge fields for a query.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Create a query.

Steps

1. Click the Query tab > Datasource sub tab.

   The Create Datasource page appears.

2. From the Query Name list, select a query.

3. Click **Create**.

   The data source file is created and the Create Datasource page refreshes. The data source file is a text file with the column headings or aliases.

4. Use the Datasource file link to download the data source file.

5. To open the data source file, right-click the **Datasource file** link and choose Open.

   The Datasource appears in a separate browser window.

6. To save the data source file, right-click the **Datasource file** link and choose Save Target As.
7. Navigate to the desired save location.
8. Optionally, modify the file name.

   The save type should be Text Document.

9. Click **Save**.

8.8 Managing Content

This chapter explains the concepts to create cover letter content types and cover letters. Sections in this topic includes:

- Cover Letter Overview
- Understanding Cover Letter Content Types
- Queries
- Creating and Managing Cover Letter Content Types
- Creating Cover Letters

8.8.1 Cover Letter Overview

A cover letter is the document that is received by the customer. It can be defined as an outbound content sent to consumers about an offer. It is used to communicate about an offer, information or a message. Typically, a cover letter is created by marketers. It can contain attachments such as data sheets, white papers, analyst reports, press releases, and so on. The cover letters are fulfilled through e-mail, fax, and print channels.

The content is stored in Oracle Content Manager (OCM) central repository. The advantages of using OCM are that it:

- Ensures that the message is consistent, when executed through all channels of execution.
- Helps organizations to maintain the integrity of the content and keep track of its usage.

- Allows organizations to maintain control over the content by giving users the access permissions on folders, by creating content type groups, and by having approval process to approve the content.
- Restricts users to access the content by having access permissions on folders, creating content type groups, and approval process.

8.8.2 Understanding Cover Letter Content Types

A cover letter content type is like a setup template that you can use while creating a content. It provides best practices that enable you to create standardized content. It contains detailed information such as attributes, data types, and blocks that are required to create a cover letter. To enable this, few seeded content types are available. Only administrators can create content types.

While creating content types, you can:

- Restrict the content type to be made available to only certain groups by mapping them to content type groups such as e-mail, print, and fax.
- Decide to set default attributes for the cover letters.
- Set the attributes so they cannot be updated.
- Specify the style sheet to be mapped to the content type.
- Associate multiple style sheets to a content type. This feature enables users to view the content or attachments using any style sheet that is associated with it. Also, a default style sheet can be mapped to the content type.

Other users can duplicate the seeded content types and modify their structures by updating the attributes.

Notes:

- If the content type is used to create a cover letter or is in In Progress status, you cannot update the content type.
- To create cover letters and content type groups that are relevant to cover letters, you must use the options that are provided with Cover Letters and not the generic OCM UI.
- only from this UI. Also, to map the content types to different groups, you must use this UI and not the OCM Content Type Groups UI.

Content Type Summary Page

In the Content Type Summary page, you can choose to view all, seeded, or inactive content types. The results are displayed in a tabular format listing the content type details such as the owner's name, its status, and so on.

8.8.2.1 Cover Letter Content Type Attributes

To create new content types, you can either copy seeded content types or create new ones based on the following attributes:

General

In this region, enter the name of the content type, select the status, and the content type group. Only if the status of the content type is active, you can select it to create cover letters. You can also select a default style sheet for the cover letter.

For more information on style sheets, see *Oracle Content Manager Implementation Guide*.

Header, Content Block, and Footer

You can use these regions to provide content in text format and images. Typically, the header and footer regions are used to provide corporate address, copyrights, contact information, terms and conditions, company logo, and so on. You can also define a default cover letter and set it so it cannot be updated.

If these regions are enabled, then while creating cover letters, users can select a live and approved version of the content from the repository based on their permissions.

Body

You can enter the HTML text for e-mails in this region. For non-HTML compliant users, you can display the content in text format. If the content type has a query associated with it, then you can choose to embed the merge fields. You can select one or more merge fields from the associated query to embed in the cover letter. These are written in SQL (Structured Query Language) format. Query items are stored in Oracle Content Manager (OCM) as cover letters of type Query.

Typically, queries are created and modified by the Marketing Administrators. For more information on Queries, see *Oracle Marketing User Guide*.

Attachments

You can send attachments of any file type with e-mails and faxes. But only .rtf and .pdf file attachments can be printed.

In e-mails, you can send binary files as attachments. You can specify the attachment details in this region. If this region is enabled, then while creating cover letters users can search the attachments based on cover letter name, description, and that are associated with Products or Offers. The search returns all cover letters that are live and approved from the repository, based on the permissions that you have.

8.8.3 Queries

Queries are Structured Query Language (SQL) statements that generate selected data from a database. A query determines which records meet the defined criteria, and returns the matched records.

Queries are the source for the mail merge fields that can be embedded in the cover letter body. They are used to generate personalized tags for mail merge, which are fulfilled by Oracle One-to-One Fulfillment, at the time of execution. The Queries page lists all the queries. You can select an existing query to view the details or click the Create button to create a new query.

For details on queries, see *Oracle Marketing User Guide*.

8.8.3.1 Creating and Updating Queries

Use the following procedure to create and update queries.

Prerequisites

None

Steps

1. Login with the One-to-One Fulfillment Administrator responsibility and navigate to Content > Query.
2. Click **Create**.

   The Create Query page appears.
3. In the Name field, enter a unique name for the query
4. Optionally, in the Description field, enter a description for the query
5. In the Query SQL field, enter the SQL SELECT statement.

It is not necessary to a semi-colon (;) at the end of the SELECT statement.

6. Click **Create**.

   The query is created with a status of In Progress.

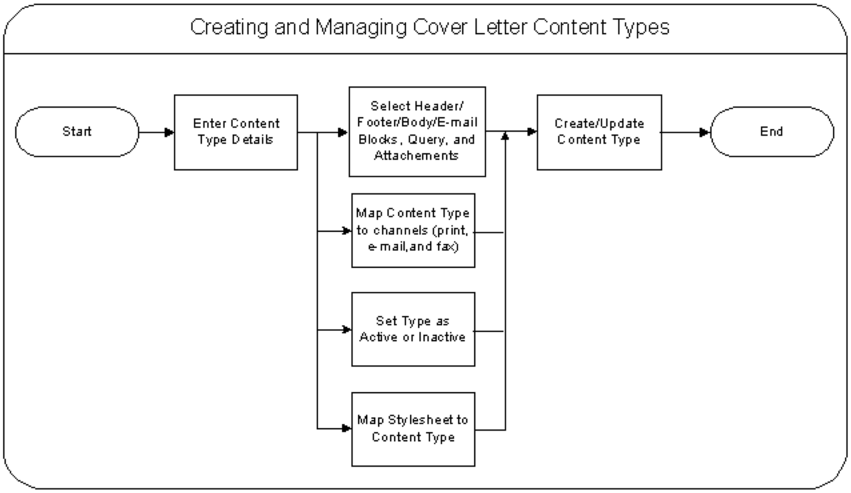
7. To update a query, select the query and click **Update**.

   - You can update the queries with status In Progress and Submit for Approval.
   - You cannot update the queries with status Approved.
   - In the Update Query page, update the required fields and click **Update**.

8.8.4 Creating and Managing Cover Letter Content Types

Only administrators can create content types. The following figure depicts the process to create a cover letter content type.

Figure 0–1 Process to Create and Manage Cover Letter Content Types


Creating and Managing Cover Letter Content Types
Start
Enter Content Type Details
Select Header/ Footer/Body/E-mail Blocks, Query, and Attachements
Map Content Type to channels (print, e-mail,and fax)
Set Type as Active or Inactive
Map Stylesheet to Content Type
Create/Update Content Type
End

The preceding figure illustrates the following process:

- Map the content type to different groups such as e-mail, fax, and Print.
- Define the requirements for the cover letters, such as header, body, e-mail blocks, queries, and footer.

- Define if the blocks must be displayed and set the default values.
- Define if the content can be updated.
- Create active content types so that they are available to create cover letters.
- Specify the default style sheet for the content type.

  The seeded style sheet (provided by OCM) is selected by default.

> **Notes:** Oracle recommends that you understand the requirements of your organization before you define the attributes for the content type. After a content type is used to create a cover letter, you cannot update it, you must first duplicate it and update the copy.

Use the following procedure to create and manage cover letter content types.

Prerequisites

None

Steps

1. Login with the One-to-One Fulfillment Administrator responsibility and navigate to Content > Content Type.
2. Click **Create**.

   The Create Content Type page appears.
3. In the General region, enter details in all the mandatory fields.
   - Select the status to be active so that it is available to create cover letters.
   - Select the groups for which the content type must be available.
   - Select a default style sheet.
4. To define Header attributes for the content type, select the appropriate Display check boxes.
   - To default the content in the Header Block and Header Image regions, select a value from the Search list.
   - Select a value for the Updateable field from the drop-down list. While creating a cover letter, this field indicates if the default value can be updated by the user or not.

- Select a value for the Required field from the drop-down list. While creating a cover letter, this field indicates if this field must be a mandatory field.

5. To select a query for the mail merge in the body of the cover letter, select a query from the Search list.
6. To define a content block in the Content Blocks region, select a value from the Search list.
   - To default the content in the Content Block, select a value from the Search list.
   - Select a value for the Updateable field from the drop-down list. While creating a cover letter, this field indicates if the default value can be updated by the user or not. This is applicable for all the fields in other regions also.
   - Select a value for the Required field from the drop-down list. While creating a cover letter, this field indicates if this field must be a mandatory field. This is applicable for all the fields in other regions also.
7. To define Footer attributes for the Content Type, select the appropriate Display check boxes.
   - To default a content in the Footer Block and Footer Image regions, select a value from the Search list.
   - Select a value for the Updateable field from the drop-down list.
   - Select a value for the Required field from the drop-down list.
8. To define an attachment in the Attachments region, select a file from the Search list.
   - Select a value for the Updateable field from the drop-down list.
   - Select a value for the Required field from the drop-down list.
9. Optionally, to preview the selected attribute, click the Preview icon.
10. Click **Save**.

    The cover letter content type is created.
11. To update a content type:
    a. Select the content type and click **Update**.

       The Create Content Type page appears displaying the content type details.

    b. Update the required fields and click **Apply**.

The content type is updated.

12. To duplicate a content type:

    a. Select the content type and click **Duplicate**.

    A duplicate copy of the content type is created.

13. To delete a content type:

    a. Select the content type and click **Delete**.

    The content type is deleted.

8.8.5 Creating Cover Letters

To create cover letter contents you must:

1. Select a cover letter content type from the Create list and click the Go button.
2. Provide content details.

> **Note:** The content type, which you select determines the content details that you must provide.

Selecting a Content Type

You must select a content type from the Cover Letter Summary page. In this page, select a content type from the Create list and click the Go button. In this page you also:

- Sort content types based on the Content Type Group.
- List all the cover letters that are created using a particular content type.
- Search cover letters based on the owner, name, location, last update date, version, and description.

Cover letter contents are stored in folders. For more information on folders and permissions, see *Oracle Content Manager Implementation Guide*.

Use the following procedure to create cover letters.

Prerequisites

A content type must exist.

Steps

1. Login with the One-to-One Fulfillment Administrator responsibility and navigate to Content > Cover Letter.
2. Select a content type from the Create list and click **Go**.

   The Create Content page appears.
3. Enter name, description, owner, and location details in the respective fields.
4. In the Header region, select the values for content block and content image from the Search list.
   - This region is enabled only if it is defined while creating the selected content type. You can select cover letters of type Image or Content Blocks from a list of values. The search result lists all the cover letters that you have access, which are live versions.
   - If this region is enabled, then you can select images or content blocks for the body header.
   - While defining the content type, if the fields are defined as non-updatable fields, then you cannot update them. You can only view the defined values.
   - To add more content blocks and images, click **Add another Row**. The order in which you add the contents are maintained during execution.
5. In the Body region, enter the text for the e-mail and fax body.
   - You can enter the content for the e-mail in an HTML format. If the e-mail is for non-html clients, you can choose to display the e-mail content in text format by clicking **Edit Text Version.** The content created for Fax groups will display the content in HTML format.
   - If the content type has any query associated with it, you can insert mail merge fields based on the available fields, either individually or all at once.
   - Users can insert mail merge fields based on the available fields.
   - To add the query mail merge fields individually, select the query and click <. Optionally, to add all the queries for mail merge, click the double < symbol.
6. In the Content Blocks region, select a content block from the Search list.

   You can add another content block in the Content Block region by clicking the Add Another Row button and selecting a content block from the Search list.

7. In the Footer region, select the values for content block and content image from the Search list.

   You can add another content block in the Footer region by clicking the Add Another Row button and selecting a content block from the Search list.

8. In the Attachments region, select the file for the attachment from the Search list. All the files in the repository are displayed.

   - You can search for attachments that are associated with Products or Offers, which are part of the campaign schedule. The search result lists all the cover letters that you have access and which are live versions.
   - You can also upload attachments from the desktop. By default, the attachments are saved in the same folder as the cover letter. You can select a different folder for which you have the Create Item permission.
   - You can add another content block by clicking the Add Another Row button and selecting a content block from the Search list.

9. In the Content Renditions region, you can upload .pdf file for e-mail content.

   - This region is enabled only for Print groups.
   - You can upload .pdf or .rtf files for as the print renditions for cover letters. The .pdf or .rtf files may contain merge fields which are related to the query associated with the cover letter.

10. Optionally, click **Save for Later**.

    The cover letter is created with the status In Progress. This option enables you to create an item, and update it later.

11. Optionally, to preview the content, click **Preview.**

    - You can preview a content, only if you have selected the **Save for Later** option.
    - You can also preview the content in text format.

12. Click **Submit for Approval**.

    The cover letter is submitted for approval.

8.8.5.1 About Submitting for Approvals

After submitting the cover letter content for approval, either one of the following happens:

- The cover letter is submitted for approval based on the status of the cover letter or any of the attachments.
- If any of the attachments are in Rework status or In Progress status, you cannot submit the cover letter content for approval.
- If all the attachments are approved, then the content goes through OCM Content Approval process. The status of the cover letter content changes to Submitted. You cannot update the content in this status.
- A cover letter is approved only after all its components are approved.

8.8.5.2 About Searching for Cover Letters

In the Cover Letter Summary page, you can narrow the search based on the content type groups, that is, e-mail, fax, and print. You can also search for cover letters based on the parameters such as cover letter name, description, folder location, and last update

The search returns all live versions of the cover letters for which you have read access permissions. The live versions are the latest approved versions of the item. There may be later versions of the item, which are In Progress status. You cannot update such items.

> **Note:** You must not edit the cover letter content types and cover letters in the generic OCM UI as the attributes may not be displayed correctly.

8.9 Managing Fulfillment Requests

Use the Status sub tab to view the status of fulfillment requests that have not been completely processed.

Tasks

You can perform the following tasks:

- Viewing the Status of a Fulfillment Request
- Canceling a Fulfillment Request
- Pausing or Resuming a Fulfillment Request

8.9.1 Viewing the Status of a Fulfillment Request

Use the following procedure to view the status of a fulfillment request.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Status tab.
2. Click the Status sub tab.

   The Request Status page appears.
3. From the View list, select a view type.

   You have the following options:

   - Submitted: Requests that have been submitted to the Advanced Queue but not yet picked up by the Fulfillment Server.
   - Re-submitted: Original Fulfillment requests that have been resubmitted to the Advanced Queue.
   - In Process: Requests being processed.
   - Previewed: This option is no longer in use.
4. Click a request identifier link.

   The View Status page appears.

8.9.2 Canceling a Fulfillment Request

Use this procedure to cancel a fulfillment request that is in the queue or is being processed.

> **Note:** With the exception of mass requests, when the Fulfillment Server retrieves a fulfillment request from the request queue, the fulfillment request cannot be canceled.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Status tab > Status sub tab.

   The Request Status page appears.

2. From the View list, select All, Submitted, Re-Submitted, or In Progress.

3. In the Action column of the request, click **Cancel**.

   The page refreshes and the canceled request is no longer available.

   To view the canceled request, see Managing Fulfillment Request History.

8.9.3 Pausing or Resuming a Fulfillment Request

Use this procedure to pause or resume a fulfillment request that is in progress.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Status tab > Status sub tab.

   The Request Status page appears.

2. From the View list, select **In Progress**.

3. In the Pause/Resume column, do one of the following:

   - To temporarily stop a request, click **Pause**.
   - To continue a paused request, click **Resume**.

   To view any jobs in the request that have already been processed, see Managing Fulfillment Request History.

8.10 Managing Fulfillment Request History

Use the History sub tab to view the history of fulfillment requests that have been processed, successfully or unsuccessfully, by a Fulfillment Server.

Tasks

You can perform the following tasks:

- Viewing the Details of a Fulfillment Request
- Viewing the Details of a Fulfillment Request
- Viewing Failed Fulfillment Requests
- Viewing a Partially Successful Fulfillment Request
- Correcting a Malformed E-Mail Job
- Viewing Job Details
- Resubmitting a Fulfillment Request
- Resubmitting a Job
- Deleting a History Entry
- Searching for a Fulfillment Request

8.10.1 Viewing the Details of a Fulfillment Request

Use this procedure to view the details of a fulfillment request.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Status tab > History sub tab.

   The Request Summary page appears.

2. From the View list, select which status of requests to view.

   - All
   - Cancelled
   - Failed
   - Partially Successful
   - Stalled
   - Successful

   The page refreshes and lists all requests of the selected status.

     - Request Identifier: Request identification number
     - Server Name: Name of the server that processed the request
     - Agent: Name of the agent who submitted the request
     - Submitted: Date the request was submitted
     - Completed: Date the request was completed
     - Outcome: Outcome of the request: Successful, Failed, Partially Successful, Cancelled, and Stalled
     - Parent Request Identifier: Request identification number and link to the original (parent) request
     - Request Type: Single or Mass
     - Channel: Email, Print, Fax, or Multi (such as fax and e-mail)

The Statistics group lists the following details.

- Fax
  - Sent: Number of fax messages sent
  - Not Sent: Number of fax messages not sent
- Print
  - Sent: Number of print items sent to the printer
  - Not Sent: Number of print items not sent to the printer
- Email:
  - Sent: Number of e-mail messages sent
  - Opened: Number of e-mail messages opened by the recipient. (The Count is displayed only if the relevant profile, Email Tracking is enabled.)
  - Unopened: Number of e-mail messages not yet opened by the recipient
  - Not Sent: E-mail message prepared but not sent
  - Malformed: Number of e-mail messages with invalid addresses
  - Do not contact: Number of e-mail messages whose recipients are flagged "Do not contact" in the Customer Database
  - Other Failures: Number of e-mail messages that could not be sent because of other reasons
  - Bounced: Number of bounced e-mail messages received for this request
  - Unsubscribed: Number of e-mail recipients who chose to unsubscribe from the e-mail recipients list
  - Resubmitted Malformed: Number of malformed e-mail messages that have been corrected and reprocessed
- The Content Details group lists information about the content:
  - Content ID
  - Content Number, the total number of content items. For example, a cover letter plus an attachment would be listed in two separate rows in the Content Details table. Content Number 1 would be the main content (the HTML or RTF form document) and Content Number 2 would be the static attachment.

- Content Name, with a link to view the content, such as an HTML or RTF file
- Content Type, such as a query, HTML rendition, RTF rendition, collateral or data
- Store, such as Oracle MarketView, Marketing Encyclopedia System, or Oracle *i*Content (Oracle Content Manager)

3. Click the number of a Request Identifier.

   The Request Details page appears.

8.10.2 Viewing Failed Fulfillment Requests

Use this procedure to see a list of failed requests.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Select the Status tab > History sub tab.

   The Request Summary page appears.

2. From the View list, select **Failed**.

   The page refreshes and lists the requests marked as failed in the Outcome column.

3. Click a Request Identifier number.

   The Request Details page appears. The Failure Details group shows the Outcome Description.

8.10.3 Viewing a Partially Successful Fulfillment Request

Use this procedure to see a partially successful fulfillment request.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Select the Status tab > History sub tab.

   The Request Summary page appears.

2. From the View list, select **Partially Successful**.

   The page refreshes and lists the partially successful requests.

3. Click the number of a Request Identifier.

   The Request Details page appears.

4. In the Statistics group, click the linked number of the Not Sent items.

   The Failures page appears, listing each failed job and the reason for failure.

8.10.4 Correcting a Malformed E-Mail Job

Use this procedure to correct an improperly addressed e-mail request.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Select the Status tab > History sub tab.

The Request Summary page appears.

2. Click the Request Identifier number of the malformed e-mail request.

   The Request Details page appears.

3. In the Statistics group, click the number that corresponds to Malformed.

   The Resubmit Malformed Email Jobs page appears. The incorrect address is in the Email Address column.

4. In the Correct Email Address field, enter the e-mail address.

5. Click **Resubmit**.

   The Request Summary page appears.

   A new Fulfillment request will be submitted for each corrected e-mail that is resubmitted. The original request identifier appears in the Parent Request Identifier of the Request Summary page.

8.10.5 Viewing Job Details

Use this procedure to see job details, such as customer information, content, and failure details.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Submit at least one job.

Steps

1. Select the Status tab > History sub tab.

   The Request Summary page appears.

2. Click the number of the Request Identifier.

   The Request Details page appears.

3. In the Job Details group, click the number of the Job.

The Job Details page appears, listing the following information groups.

- Customer Information:
  - Customer Identifier
  - Customer Name
  - Email Address: E-mail address of the recipient
  - Fax Number
  - Printer Address: Name of the printer to which the request was sent
- Content Details:
  - Content ID
  - Content Number, the total number of content items. For example, a cover letter plus an attachment would be listed in two separate rows in the Content Details table. Content Number 1 would be the main content (the HTML or RTF form document) and Content Number 2 would be the static attachment.
  - Content Name, with a link to view the content, such as an HTML or RTF file
  - Content Type, such as a query, HTML rendition, RTF rendition, collateral or data
  - Store, such as Oracle MarketView, Marketing Encyclopedia System, or Oracle *i*Content (Oracle Content Manager)
- Failure Details (Only applicable if the request was partially successful):
  - Channel: Email, Fax, and Print
  - Address: Email address, fax number, or printer
  - Reason: Cause for the failure

8.10.6 Resubmitting a Job

Use this procedure to resubmit a job that was sent unsuccessfully.

Note: For Failed requests, if the Request Details page does not list Job Details, the request has failed before individual jobs could be specified. In such cases, it is not possible to resubmit jobs because none have been created.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

Submit at least one job.

Steps

1. Select the Status tab > History sub tab.

   The Request Summary page appears.

2. Click the Request Identifier number of the malformed e-mail request.

   The Request Details page appears.

3. In the Job Details group, click the linked number of the Job.

   The Job Details page appears.

4. Click **Resubmit**.

   The Resubmit Job page appears.

5. From the Channel list, select Email, Fax, or Print.
6. In the Address field, enter the email address or fax number of the recipient, or enter the printer name.
7. Click **Resubmit**.

8.10.7 Resubmitting a Fulfillment Request

Use this procedure to resubmit a failed fulfillment request.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Status tab > History sub tab.

   The Request History page appears.

2. In the Action column, click **Submit**.

8.10.8 Deleting a History Entry

Use this procedure to delete a history entry for a fulfillment request.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Status tab > History sub tab.

   The Request Summary page appears.

2. Optionally, from the View list, select which status of requests to view.

   - All

- Cancelled
- Failed
- Partially Successful
- Stalled
- Successful

The page refreshes and lists all requests of the selected status.

3. Click **Select** for the history entry that you want to delete.
4. Click **Delete**.

   The selected history entry is removed from the list and the Request History page refreshes.

8.11 Searching for a Fulfillment Request

Use the Search sub tab to search for a fulfillment request that has been processed by a Fulfillment Server.

Use this procedure to search for a fulfillment request.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Click the Status tab > Search sub tab.

   The Search page appears.

2. From the list, select the search type:

   - Customer Name
   - Customer Identifier

- Request Identifier

3. In the field, type the value for which to search.
4. Click **Search**.

   The list of search results displays the following information about each fulfillment request.

   - The results display a summary of the Customer Identifier, Customer Name, Request Identifier, Job, and Job Outcome. Users can click the History Request Identifier number to view the Request Details, which displays a summary of the request (server name, submitted & completed times, agent who initiated the request, and the outcome code, statistics, content, and job summary).
   - Users can click the job number to view the job details, which displays a summary of the job (customer information, content details, and any applicable failure details).

8.12 Using the Quick Find Feature

Use the Quick Find feature to find a Fulfillment-related item, such as a server or template.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

Prerequisites

None

Steps

1. Select any tab in the administration console.
2. From the Quick Find list, select the search type:
   - Group (fulfillment agents)
   - Master Document
   - Query

- Server (1)
- Status
- Template

3. In the Quick Find field, enter the search criterion. The criteria are case sensitive, except for Request Status which must be typed in all capital letters. The Request Status types are:

   (Type capital letters only):

   * SUBMITTED
   * RESUBMITTED
   * PAUSED
   * PREVIEWED
   * RESUMED
   * CANCELLED

   You can search for an item by using character strings and the wildcard symbol (%). For example, to find servers starting with A, type "A%".

4. Click **Go**.

   The Search Results page appears.

5. To view the details of an item in the search results, click the item link.

   The Details page appears.

8.13 Using Merge Fields

The Fulfillment Server code uses sets of merge field delimiters to determine whether a document contains merge fields. It searches for the left delimiter of each pair within the document text. When it finds a left delimiter, it tries to then pair it with a right delimiter. If it does not find the right delimiter, the next pair of delimiters is searched for, and so on, until either a complete merge field is found or the list of merge delimiters is exhausted.

Topics include:

- Default and Wildcard Merge Delimiter Pairs
- Different Representations of the Same Character

- Using Encrypted Merge Fields
- Creating an RTF Form Document with Merge Fields
- Creating a PDF Form Document with Merge Fields

8.13.1 Default and Wildcard Merge Delimiter Pairs

There are three sets of merge delimiters, the first of which is a wildcard pair. This first set can be overridden with the Java property flags 'merge.LeftDelimiterString' and 'merge.RightDelimiterString' (Java properties must be preceded with '-D' when part of the java command). You can also use the profile options JTF_FM_LMERGE_DELIM and JTF_FM_RMERGE_DELIM to select from a list of seeded merge delimiters. The other two sets are static and therefore always a part of the list. The list includes:

Left Delimeter	Right Delimeter	Description
&laquo;	&raquo;	ASCII alphabetic delimiter codes. These are the defaults for the wildcard pair.
&#171;	&#187;	ASCII numeric delimiter codes.
'\u00AB'	'\u00BB'	Unicode delimiter codes.

Some of these delimiters begin with an ampersand '&' and end with a semi-colon ';'. These are special characters that are recognized by HTML. In the case of '&#171;', it is interpreted by HTML as a character having the decimal numeric value of 171.

8.13.2 Different Representations of the Same Character

The Fulfillment Server allows multiple merge delimiters due to the way these characters are represented. Unicode characters are represented in hexadecimal values. HTML uses decimal values. So the left-pointing double angle quotation mark (guillemet) can be represented in decimal as "&#171;" or in Hexadecimal as '\u00AB' or even in Octal as '\253'. The Octal representation in Unicode is '\u00AB'. Some log files contain '\253' and '\273'. The Fulfillment Server needs to have each of these representations because it parses characters and character sequences. When you see the character a left-pointing double angle bracket in your Web browser, it might be represented by the character sequence `&#171;` in the HTML source. The fulfillment server parses this source and therefore parses the sequence '&' followed by '#' followed by '1' and '7' and '1' and ';'. The same is true for any character sequence set with the flags ‘merge.LeftDelimiterString’ and ‘merge.RightDelimiterString’.

The following table lists the different numeric character conversions of the default merge delimiters that are used by the Fulfillment Server.

Character	Decimal	Hexadecimal	Octal
left-pointing double < bracket (guillemet)	171	AB	253
right-pointing double > bracket (guillemet)	187	BB	273

8.13.3 Using Encrypted Merge Fields

Oracle One-to-One Fulfillment has encrypted merge fields. When an encrypted merge field is found, the server encrypts the merge value and then leaves the delimiters in place so that it can later be recognized and decrypted. The default string delimiters are:

- Left delimiter string “*encrypt(“
- Right delimiter string “)*”

You can override these strings with the Java property flags ‘merge.LeftEncryptDelimiter’ and ‘merge.RightEncryptDelimiter’. You can also use the profile options JTF_FM_LENCRYPT_DELIM and JTF_FM_RENCRYPT_DELIM which only allow you to select from a list of seeded values.

8.13.4 Creating an RTF Form Document with Merge Fields

Use the following procedure to create Rich Text Format (RTF) form documents, containing merge fields, to use with the Oracle One-to-One Fulfillment native print function.

Login

Not applicable

Responsibility

Not applicable

Prerequisites

None

Steps

1. In Microsoft Word, open a Word document. The document must have all necessary text, merge field references, and images.
2. Put the cursor on the highest Word toolbar, and click to open the mouse menu.
3. Select **Forms**.

   The Forms toolbar appears.
4. Highlight the first text that you want to be a merge field.
5. On the Forms toolbar, click **ab|** (Text Form Field).

   The text is highlighted.
6. Double click the highlighted text.

   The Text Form Field Options box appears.
7. In the Default text field, enter the display name of the form field.
8. Click **Add Help Text**.

   The Form Field Help Text box appears.
9. In the text field, enter the name of the merge field. The merge field name must be exactly the same the name or alias that is used in the SQL query and must be enclosed within "<?" and " ?>", for example, <? FIRST_NAME ?>. The merge delimiters must always be in upper case.
10. Click **OK**.

    The Form Field Help Text box closes.
11. In the Text Form Field Options box, click **OK**.

    The Text Form Field Options box closes. The highlighted text form field name appears.
12. For each merge field, repeat steps 4 through 11.
13. Save the document as a Rich Text Format file with an rtf file extension.

Creating a Table with a Non-Repeating Row

The following table is an example of a table containing a non-repeating row.

Contact	Email Address	Status of Email	Date Processed
NAME	EMAIL_ADDRESS	STATUS	SENT_DATE

To include non-repeating data within an RTF document, use the following procedure.

Steps

1. Insert a table, selecting the appropriate number of columns and one or two rows.
2. If required, enter the appropriate column header names.
3. Insert a text form field into the first cell of the second row (assuming there are column headers in the first row). To insert merge fields, see steps 4 through 11 above.
4. Repeat until there is a text form field in each cell of the row.

   The following table is an example of how the table containing a non repeating row would appear in the printed document.

Contact	Email Address	Status	Date Processed
CHERYL LAWRENCE	Cheryl.Lawrence@vision.com	SENT	21-JAN-2003

Creating a Table with Repeating Rows

A table with repeating rows of data enables multiple rows of data to be returned by a SQL query and presented in a table. The following table is an example of a table that contains repeating rows.

Contact	Email Address	Status of Email	Date Processed
row repeat NAME	EMAIL_ADDRESS	STATUS	SENT_DATE row repeat

Steps

1. Insert a table, selecting the appropriate number of columns and one or two rows.

2. If required, enter the appropriate column header names.
3. Insert a text form field into the first cell of the second row. To insert merge fields, see steps 4 through 11 above
4. Repeat until there is a text form field in each cell of the row.
5. Add a new form field at the start of the second row (assuming there are column headers in the first row).
6. On the Forms toolbar, click **ab|** (Text Form Field).

   The text is highlighted.
7. Double click the highlighted text.

   The Text Form Field Options box appears.
8. In the Default text field, enter the display name of the form field, for example "row repeat".
9. Click Add Help Text.

   The Form Field Help Text box appears.
10. In the text field, use the for-each:<group name> command (surrounded by the <? and ?> delimiters) to indicate the start of the repeating group (in this case, a row), for example;, <?for-each:row?>.
11. Click **OK**.

    The Form Field Help Text box closes.
12. In the Text Form Field Options box, click **OK**.

    The Text Form Field Options box closes. In the Word document, the highlighted text form field name appears.
13. Add a new form field at the end of the second row (assuming there are column headers in the first row).
14. On the Forms toolbar, click ab| (Text Form Field).

    The text is highlighted.
15. Double click the highlighted text.

    The Text Form Field Options box appears.

    In the Default text field, enter the same display name for the form field that was used at the start of the row, for example "row repeat"

16. Click **Add Help Text**.

    The Form Field Help Text box appears.

17. In the text field, use the end for-each command (surrounded by the <? and ?> delimiters) to indicate the end of the repeating group (in this case, a row), for example;, <?end for-each?>.

18. Click **OK**.

    The Form Field Help Text box closes.

19. In the Text Form Field Options box, click **OK**.

    The Text Form Field Options box closes. The highlighted text form field name appears.

    The following table is an example of how the table containing repeating rows would appear in the printed document.

Contact	Email Address	Status	Date Processed
CHERYL LAWRENCE	Cheryl.Lawrence@vision.com	SENT	21-JAN-2003
CHERYL LAWRENCE	Cheryl.Lawrence@vision.com	SENT	29-JAN-2003
BRYAN WEBB	BRYAN.WEBB@vision.com	SENT	21-jan-2003
MICHAEL Z ROBERTSON	MICHAEL.ROBERTSON@vision.com	NOT SENT	21-jan-2003
MICHAEL Z ROBERTSON	MICHAEL.ROBERTSON@vision.com	SENT	28-jan-2003

Creating a Repeating Table

A repeating table can have multiple tables of data returned by a SQL query. The following table is an example of repeating rows.

```
mytable group
```

Contact	Email Address	Status	Date
NAME	EMAIL_ADDRESS	STATUS	SENT_DATE

```
mytable group
```

Steps

1. Insert a table, selecting the appropriate number of columns and one or two rows.
2. If required, enter the appropriate column header names.
3. Insert a text form field into the first cell of the second row, by following steps 4 through 11 above.
4. Repeat until there is a text form field in each cell of the row.
5. Add a new form field just above the table .
6. On the Forms toolbar, click **ab|** (Text Form Field).

   The text is highlighted.
7. Double click the highlighted text.

   The Text Form Field Options box appears.
8. In the Default text field, enter the display name of the form field, for example "mytable group".
9. Click **Add Help Text**.

   The Form Field Help Text box appears.
10. In the text field, use the `for-each:<group name>` command (surrounded by the <? and ?> delimiters) to indicate the start of the repeating group (in this case, a table), for example;, `<?for-each:mytable?>`.
11. Click **OK**.

    The Form Field Help Text box closes.
12. In the Text Form Field Options box, click **OK**.

    The Text Form Field Options box closes. In the Word document, the highlighted text form field name appears.
13. Add a new form field just after the table.
14. On the Forms toolbar, click **ab|** (Text Form Field).

    The text is highlighted.
15. Double click the highlighted text.

    The Text Form Field Options box appears.

16. In the Default text field, enter the same display name for the form field that was used at the start of the row, for example "mytable group".

17. Click **Add Help Text**.

    The Form Field Help Text box appears.

18. In the text field, use the `end for-each` command (surrounded by the <? and ?> delimiters) to indicate the end of the repeating group (in this case, a table), for example, `<?end for-each?>`.

19. Click **OK**.

    The Form Field Help Text box closes.

20. In the Text Form Field Options box, click **OK**.

    The Text Form Field Options box closes. In the Word document, the highlighted text form field name appears.

    The following example shows how repeating tables would appear in a printed document.

Contact	Email Address	Status	Date
CHERYL LAWRENCE	Cheryl.Lawrence@vision.com	SENT	21-JAN-2003

Contact	Email Address	Status	Date
CHERYL LAWRENCE	Cheryl.Lawrence@vision.com	SENT	29-JAN-2003

Contact	Email Address	Status	Date
BRYAN WEBB	BRYAN.WEBB@vision.com	SENT	21-jan-2003

Contact	Email Address	Status	Date
MICHAEL Z ROBERTSON	MICHAEL.ROBERTSON@vision.com	NOT SENT	21-jan-2003

Contact	Email Address	Status	Date
MICHAEL Z ROBERTSON	MICHAEL.ROBERTSON@vision.com	SENT	28-jan-2003

The following examples are the repeating and non-repeating data scenarios that are supported by the RTF print feature:

- If "for-each" and "end for-each" appears in normal text, then the text between the tags/form fields will be repeated.
- If "for-each" and "end for-each" appears at a table's or multiple tables' start and end points, then the included table(s) will be repeated.
- If "for-each" and "end for-each" appears in a single table cell, then the text between the tags/form fields in the table cell will be repeated.
- If "for-each" and "end for-each" appears in two different cells but on the same row, then the single row will be repeated.
- If "for-each" and "end for-each" appears in two different table rows, then the rows between the tags/form fields will be repeated.

Conditions

The following examples show conditions with tables or text to enable information to be displayed *only* if the condition is true.

Example 8–1 Table Reference is False

```
if references is false
```

Reference Document	Document Type	Date Required
row repeat DOC_NAME	DOC_TYPE	DATE_REQD row repeat

```
end if
```

Example 8–2 Text Reference is False

```
if references is false
REQD_REF_TEXT
end if
```

Use the following procedure to display a table if a particular condition is true.

1. Insert a table, selecting the appropriate number of columns and rows.
2. If required, enter the appropriate column header names.
3. Insert a text form field into the first cell of the second row. To insert merge fields, see steps 4 through 11 above. Repeat until there is a text form field in each cell of the row.
4. Add a new form field just above the table or text.
5. On the Forms toolbar, click **ab|** (Text Form Field).

   The text is highlighted.
6. Double click the highlighted text.

The Text Form Field Options box appears.

7. In the Default text field, enter the display name of the form field, for example "if references is false".
8. Click **Add Help Text**.

   The Form Field Help Text box appears.
9. In the text field, use following condition command format to evaluate when the condition is true: `<?if:<merge field><condition>'<value>'?>`

   for example, `<?if:reference='false'?>`.
10. Click **OK**.

    The Form Field Help Text box closes.
11. In the Text Form Field Options box, click **OK**.

    The Text Form Field Options box closes. In the Word document, the highlighted text form field name appears.
12. Add a new form field just after the table or text.
13. On the Forms toolbar, click **ab|** (Text Form Field).

    The text is highlighted.
14. Double click the highlighted text.

    The Text Form Field Options box appears.
15. In the Default text field, enter the same display name for the form field that was used at the start of the row, for example "end if".
16. Click **Add Help Text**.

The Form Field Help Text box appears.

17. In the text field, use the end if command (surrounded by the <? and ?> delimiters) to indicate the end of the conditional text or table, for example, <?end if?>.

18. Click **OK**.

    The Form Field Help Text box closes.

19. In the Text Form Field Options box, click **OK**.

    The Text Form Field Options box closes. In the Word document, the highlighted text form field name appears.

The following examples show condition commands used around tables or text.

Example 8–3 Table if Condition is True

The following example shows how the table will appear if the condition is true.

Reference Document	Document Type	Date Required
United Kingdom Student Visa	Copy	5/18/04
United States Passport	Copy	5/18/04
High School Certificate	Original	3/27/04
Degree Certificate	Original	3/27/04

Example 8–4 Text if Condition is True

The following example shows how the text will appear if the condition is true.

> Your application for the Bachelor of Arts in Fine Arts, Full Time study at the Belmont Campus requires that your Transcripts be submitted by June 1st, 2004. Please submit this item directly to the Admissions Office.

8.13.5 Creating a PDF Form Document with Merge Fields

Use the following procedure to create merge fields in an Adobe Acrobat PDF form document. Oracle One-to-One Fulfillment supports only Adobe Acrobat Version 5.

Login

Not applicable

Responsibility

Not applicable

Prerequisites

None

Steps

1. Do one of the following:
   - If the document is already a PDF file, then proceed to step 4.
   - If the document is a Microsoft Word file, then proceed to step 2.
2. In Microsoft Word, open the document that you want to use as a Fulfillment template. The document must not contain merge fields.
3. In the Word menu, select **Acrobat** or **Adobe PDF** > **Convert to PDF**.

   The Acrobat progress indicator shows the status of the conversion of the Word file to a PDF file.
4. In Adobe Acrobat, open the PDF file.
5. On the Acrobat toolbar, click the Form Tool.
6. Place the cursor where you want create the first merge field, then click and drag the Form Tool to the size that you want the merge field to be.

   The Field Properties box opens.
7. In the Name field, enter the name of the merge field. The name must be exactly the same as the form field name or alias in the SQL query, such as FIRST_NAME. The merge delimiters must always be in upper case.
8. In the Appearance tab, select the text options for this field.
9. In the Options tab, select the text alignment.
10. Click **OK**.
11. For each merge field that you want to add, repeat steps 3 through 10.
12. Save the file.

8.14 Setting Up High Quality Printing

Use the following procedures to set up high quality Oracle printing.

8.14.1 Configuring the Printer

Use the following procedure to configure the printer for high quality Oracle printing.

Login

Not applicable

Responsibility

Not applicable

Prerequisites

None

Steps

1. Verify that you can print from the Fulfillment Server host. Create a test document such as a text file called hello.txt that contains simple text.
2. Identify the printer. You may need to ask the network administrator or system administrator the proper name or destination.
3. Use the proper executable to print the file hello.txt.
   - For UNIX or Linux, use the lp command.

     ```
     Example usage: lp -d loc2-prt1-rm108 hello.txt.
     ```

   - For Windows, use the print command.

     ```
     Example usage: PRINT /D:loc2-prt1-rm108 hello.txt.
     ```

4. Verify that the document printed successfully. The High Quality Printing Solution depends on printing by using the operating system command.
5. Download and install the third party tool XPDF. Depending on your operating system, you may need to download source and build an executable, although most operating systems already have images built.

> **Note:** Become familiar with the Pasta User's Guide. Under the Configuration File Options section it explains how to set up the preprocess option in the configuration file, pasta.cfg, to instruct Pasta to preprocess PDF files before sending them to the printer. Preprocessing involves running the file through the pdftops utility to convert the unprintable PDF file to the printer-recognized format Post Script.

6. Save the configuration.
7. Test the configuration. Using the same printer name that you used for the lp or print command, print a PDF file using FNDPSTAX. The –pn parameter is where you specify the printer name.

8.14.2 Setting Up a Pasta Print Server

Use the following procedure to set up a Pasta print server.

Login

HTML Login URL

Responsibility

One-to-One Fulfillment Administrator

None

Steps

1. On the Printer tab, click **Create.** to begin creating a Pasta print server. Print servers are primarily used for RightFax Printing, but are also used in High Quality Printing as a way to identify which printers are Pasta Printers.
2. In the Print Server Name field, enter a generic name such as Pasta Print Server.
3. From the Server Type list, select **Pasta**. For Pasta printers, no Connection String is necessary.
4. From the main Print Servers table, select the Pasta Printer.
5. On the Detail page, select **Create Printer** to add your Pasta printer to this print server grouping.

6. In the Printer Name field, enter a label or a user-friendly name that users will recognize. For example, if this printer is the printer installed in room 108, enter Room 108 Pasta Printer.
7. Optionally, enter a Description.
8. In the Network Name field, enter the same name that you used to print using lp or print and FNDPSTAX. This name is passed to the call to Pasta.
9. Click **Apply**.
10. If you plan to configure more than one Pasta printer, test each one using the previous steps before adding it to the Pasta Print Server that you created.

8.14.3 Verifying the Printer Configuration

After you have successfully tested and configured the printer, run a Fulfillment request. First create and upload the RTF template. When you have identified the document, submit a normal print request. For the printer address parameter, use the Network Name field from the Administration Console setup.

If you have installed the necessary 11.5.10 Fulfillment prerequisites, you will get a high quality printed version of the document that you sent to the printer. If you have not entered your printer Network Name in the Administration Console as a printer connected to a Pasta Print Server, the Fulfillment Server will behave as though the printer address is a RightFax printer. The request will fail if RightFax does not have a printer with the same name as the Network Name.

All documents submitted to a Pasta printer must be either PDF or RTF. If you have attachments such as text or HTML, the print request will fail because the Pasta printer will not accept these document types.

A

Fulfillment Server Commands and Startup Scripts

Use the commands and startup scripts in the following topics to control the Fulfillment Server.

- Section A.1, "Fulfillment Server Commands"
- Section A.2, "Fulfillment Server Startup Script Parameters"

A.1 Fulfillment Server Commands

This section describes the commands used to operate a Fulfillment Server when it is running. If the engine.CommandPromptEnabledparameter is set to true, then you can enter the command at the same command prompt from which you started the Fulfillment Server.

The following table lists and describes the Fulfillment Server commands.

Table A–1 Fulfillment Server Commands

Command	Description
flush logs	Force the log buffers to write to disk.
help	Display all of the valid commands for the Fulfillment Server.
kill	Stop the server. To restart the server, run the server startup script.
list error <n>	List a specific entry in the error log. The parameter n can be a single number, a range of numbers, or the word count to list the number of entries in the log.

Table A–1 Fulfillment Server Commands

Command	Description
list events <n>	List a specific entry in the event log. The parameter n can be a single number, a range, or the word count to list the number of entries in the log.
mstat	Print statistics about the Fulfillment Server monitors.
pstat	Print statistics about the Fulfillment Server processes.
restart	Stop and automatically restart the Fulfillment Server.
start	Start a stopped Fulfillment Server.
stat	Print statistics about the Fulfillment Server.
stop	Pause Fulfillment Server processing. To resume processing, use the start command.
tstat	Print statistics about the java thread activity in the Fulfillment Server.

A.2 Fulfillment Server Startup Script Parameters

This appendix describes the Fulfillment Server server parameters that can be set in the Fulfillment Server startup script. Parameters are listed after the java or jre command and before the name of the class that is being executed. Prefix the parameter with -D to pass runtime property settings to the Java Virtual Machine. For example,

```
-Dengine.AOLJ.config=<full_path_to_dbc_file> \
```

See Creating a Startup Script for the Fulfillment Server for more information about startup scripts.

Topics include:

- Required Flags
- Optional Flags
- Debug Flags
- Additional Java Flags
- Language Character Parsing

A.2.1 Required Flags

The flags listed in the following table are the only flags required to run a Fulfillment Server. The server will not start without them. These flags may be subject to change in future releases depending on changing system requirements.

Table A–2 Required Flags

Flag	Description
engine.ServerID	Identifies the Fulfillment Server as referenced in the database. With the server id, the Fulfillment Server can get all of it's configuration from the database without the need for any of the optional flags.
engine.AOLJ.config	Specifies the full-path location of the dbc file that the server uses to get connections to the database.

A.2.2 Optional Flags

The following flags are optional. Certain flags, such as the logging flags, can degrade the performance of the server. Other flags can improve the performance of the server.

Note: Fulfillment Server flags are added or deprecated from time to time, so that this list may change.

Table A–3 Optional Flags

Flag	Description
email.PrefersMultipartAlternative	Specifies that the server is to allow a text-only alternative for the DATA or QUERY content that is sent by e-mail. Default is 'false'.
email.TextOnly	Specifies that the server is to send out text-only e-mail instead of the HTML-based e-mail that it sends out for it's DATA or QUERY contents. Default is 'false'
engine.CommandPort	Specifies the numeric port id, on the servers' hardware, that the server will listen to for receipt of remote commands. The only other way to send a command to the server is by the command prompt from which it was started (see engine.CommandPromptEnabled). Default is to not have command port defined.

Table A–3 Optional Flags (Cont.)

Flag	Description
engine.CommandPromptEnabled	Specifies that server is to be able to receive commands by the command prompt from which it was started. This flag is useless when running the process in background on UNIX or as a service on Windows NT. For those types of servers, the flag engine.CommandPrompt should be used. The default value for this flag is 'true'.
engine.ErrorLog	Specifies the name of the log used by the server to log errors. The default is 'Errors.log'. Also see 'engine.LogPath' and 'engine.LogLevel'.
engine.EventLog	Specifies the name of the log used by the server to log events. The default is 'Events.log'. Also see 'engine.LogPath'.
engine.FaxEnabler	Indicates the class that the Fulfillment Server is to use for sending faxes. This class must implement the 'oracle.apps.jtf.fm.engine.disp.FaxEnabler' interface to be used as a FaxEnabler. Also see 'engine.PrintEnabler'. No Default value.
engine.LogLevel	Indicates the level of detail that will be written to the error log (see engine.ErrorLog). For each number from 1-n the server will write an extra layer of chained exception traces to the errors log. The exceptions are chained together so that the source of the errors as they occur can be traced.
engine.LogPath	Specifies the directory path where Fulfillment Server logging occurs. The default if this flag is not set is to use 'engine.ServerPath' or 'logs' under the directory from which the server is started if that is also not set.
engine.LogRequests	A Boolean flag that indicates that the original XML-based request pulled from and Advanced Queue by the server is to be written to disk. The requests will be saved as an XML file in the form of #####.xml where '#####' is the request id. The requests will be saved in a directory called 'requests' under the log path. Also see 'engine.LogPath' and 'engine.LogDispatches'. Default is 'false'.
engine.PrintEnabler	Indicates the class that the Fulfillment Server is to use for sending print jobs to the printer. This class must implement the 'oracle.apps.jtf.fm.engine.disp.PrintEnabler' interface to be used as a PrintEnabler. Also see 'engine.FaxEnabler'.
engine.RefreshRate	Indicates the amount of time, in milliseconds, between processor refreshes. This is when the servers' processor checks the status of all sub-processes to make sure they are functioning correctly.

Table A–3 Optional Flags (Cont.)

Flag	Description
engine.ResultSetWarningLimit	Indicates how large of a ResetSet can be returned from a database query, before a warning will be logged to the Fulfillment Servers error log. This is to warn the server administrator that in case of an OutOfMemoryError, it might have been caused by the size of the Result Set. *This is not so much a problem in the newer versions of the Fulfillment Server. Also see 'log.ShowWarnings'. Default is '100'.
db.ExitOnFailure	Specifies that the Fulfillment Server is to shutdown when the database becomes unavailable. Default is 'false'.
history.NoContents	Indicates the Fulfillment Server is not to write Content Level History. This is detailed information about the processing and delivery of Fulfillment Content. This flag should be used when content history is not needed but a performance boost is. Also see 'history.NoInteractions'. Default is 'false'
history.NoInteractions	Indicates the Fulfillment Server is not to write Interaction History. This is detailed information about the interaction between the agent and parties addressed in the Fulfillment request. This flag should be used when Interaction History is not needed but a performance boost is. Also see 'history.NoContents'. Default is 'false'.
pkg.ident	Specifies the Fulfillment Server should write class file versions of the Fulfillment Server to the events log. *This only works when the oracle.apps.jtf.fm.FulfillmentServer class is used to start the Fulfillment Server instead of oracle.apps.jtf.fm.engine.Processor. Also see pkg.ident. Default is 'false'.
pkg.showmissing	Works with the 'pkg.ident' flag and specifies the Fulfillment Server note missing class files in the events log. This is useful when wrapping an older version of the Fulfillment Server with the FulfillmentServer class. *This only works when the oracle.apps.jtf.fm.FulfillmentServer class is used to start the Fulfillment Server instead of oracle.apps.jtf.fm.engine.Processor. Also see 'pkg.ident'. Default is 'false'.
log.ShowWarnings	Indicates that warnings are to be logged to the error log. By default, warnings do not show up in the error log. If this flag is not set, then the first time a warning appears, a, one time only, message shows up in the events log signifying that a warning had been issued. Default is 'false'.
smtp.port	Overrides the default port '25' for e-mail sent to an SMTP e-mail server.

Table A–3 Optional Flags (Cont.)

Flag	Description
smtp.server	Overrides the default SMTP server that is set using the Fulfillment Administration Console.

A.2.3 Debug Flags

The Fulfillment Server can be debugged, at runtime, through the use of a debug flag in the Fulfillment Server startup script or on the command line. The debug information is written to the Fulfillment Server events log, with debug=log, or to Central Logging in the database. In some cases, through use of third-party debugging (for example, Java Mail API), debug information is written to the console.

> **Note:** Oracle advises that you do not try to use debug flags to improve server performance. Using any of these flags reduces the performance of the server by the time it takes for the debug sections to run. In most cases, debug information is either written to the Events log or it is written to the command prompt.

The correct syntax for this flag is `-Ddebug=<flag>[,<flag>],` for example:

```
-Ddebug=email,request,content, query
```

This comma-separated list allows for individually setting the debug flags that are listed in the following table.

Table A–4 Fulfillment Server Debug Flags

Flag	Description
full	Turns on all debug flags, including 'log' which disables Central Logging.
off	Turns off all debugging.
content	Causes request content, as downloaded from the database, to be written to local disk on which the server is running. The directory is called 'contents' and can be found under the log directory.
e-mail	Causes details of the delivery of e-mail to be printed to the Events log and to the console. (Not all information printed to the console is recorded in the Events log.)

Table A–4 Fulfillment Server Debug Flags (Cont.)

Flag	Description
fax	Debug the delivery of faxes.
history	Debug the writing of Fulfillment Request History, Fulfillment Content History and Interaction History
log	Debug the creation of local disk log files. This flag disables the use of Central Logging.
monitor	Debug the operations of the Fulfillment Server's internal monitors. All vital internal processes of the server are handled by these monitors.
print	Debug the sending of print jobs.
processor	Debug the functions of the server's main thread.
query	Debug queries executed by the server. These are written to the Events log.
request	Debug requests that are processed by the server. Details of the request are written to the Events log.
header	Debug extended headers. Each is written as part of the e-mail header for e-mail sent by the server. Additional details of the request are written to the header such as request id, party_id, and so on.
jdbc	Debug JDBC calls made by the server. JDBC is how Java communicates with the database.
fault	Debug the functioning of the server's fault monitor. This provides details of request fault detection and handling.
mass	Debug the processing of Mass Requests.
failure	Debug failures that occur as a normal part of processing a request. The failure is written to the Error log to record a stack trace from where the failure originated. This skews the actual error count because an exception is created to track the stack trace of the failure.
merge	Debug the parsing of merge fields and insertion of bind variables within dynamic request content (Master Documents and e-mail subject field).
queue	Debug the reading and writing to Advanced Queues (AQ). The XML-based requests and responses are written to the Events log.

Table A–4 Fulfillment Server Debug Flags (Cont.)

Flag	Description
bind	Debug the use of bind data associated with content of a fulfillment request.
labels	Debug PDF Labels requests.

A.2.4 Additional Java Flags

The standard flags of the Java programming language are useful in using the Fulfillment Server to maximum benefit. Some of these flags may not be available for all versions of Java that are used to run the Fulfillment Server. You can view a list of Java flags that work with your version of Java by typing the full path to the java or jre command, as you would use to start the Fulfillment Server.

Table A–5 Additional Java Flags

Flag	Description
ms	Specifies the amount of memory to allocate to the Fulfillment Server on startup. Use it as `-Dms##[k|m]` where ## is the amount of memory, k is kilobytes and m is megabytes. Specifying a larger value for ms at startup can produce some performance gains depending on how memory intensive the server is for the types of requests it is processing. The default value is 1 MB.
mx	Specifies the maximum amount of memory to allocate to the Fulfillment Server. The server will start out with the value for ‘ms’ and work it’s way up to the value of mx as it is needed. It is to be used as `-Dmx##[k|m]` where ## is the amount of memory and k is kilobytes and m is megabytes. This flag should be used if the server is encountering an OutOfMemoryError. The default value is 16 MB.

A.2.5 Language Character Parsing

The Fulfillment Server sometimes incorrectly parses certain natural language characters in e-mail when using default environment settings and Java runtime options. The following configuration settings in the file FM_start.sh work for some of the languages that have encoding issues with the default settings.

Chinese (Traditional)

```
export LANG=zh_TW.big5
```

```
-Dengine.CharEncoding=BIG5
```

French

```
export LANG=fr_FR.iso88591
-Dengine.CharEncoding=8859_1
```

Korean (Hangul)

```
-Dfile.encoding=EUC-KR
```

Japanese

```
export LANG=ja_JP.PCK
-Dengine.CharEncoding=SHIFT_JIS
```

Russian (Cyrillic)

```
export LANG=ru_RU.KOI8R
```

or

```
export LANG=ru_RU.iso88591
```

B APIs

Oracle One-to-One Fulfillment is an automated way to send information to customers. It provides the ability to immediately satisfy a customer's requests for information, literature, and other correspondence or collateral. Customer service representatives handle a variety of requests ranging from product and service inquiries, pricing questions, billing inquiries and general customer care issues. Many of these requests result in distribution of literature, collateral, forms of application, letters or correspondence to the customer.

Oracle One-to-One Fulfillment provides the ability for interaction center administrators, mobile field representatives, marketing managers, customer care representatives and other service agents to respond to different customer needs quickly and easily by using e-mail, fax or print channels.

PROCEDURE NAME	DESCRIPTION
Start_Request	Generates the history request identifier (x_hist_id), used as the tracking identifier for the fulfillment request..
Get_Content_XML	Forms the <item> element of the XML request. You can call this API as many times as necessary.
Submit_Request	Initiates the fulfillment request process.

B.1 Priority Levels

The following table lists the priority levels defined as global constants in package JTF_FM_REQUEST_GRP. The default is G_PRIORITY_REGULAR. Unused priority numbers are reserved for future use.

LEVEL	NUMBER
G_PRIORITY_HIGHEST	1
G_PRIORITY_SPECIALITY_FLAG	6
G_PRIORITY_REGULAR	7

B.2 Interaction History Logging

One-to-One Fulfillment requests information can be logged in Interaction History Tables in the following ways:

- Interaction History logging for Single Requests. Valid party_id should be passed to log interaction history.
- For any other type of Requests except "TEST" requests, p_per_user_history should be set to 'Y' to log interaction history. There is no interaction history logging for "TEST" requests.

B.3 Extended Headers

Extended Header information can be passed to the One-to-One Fulfillment Server using the (p_)extended_header parameter of the API. This is an optional parameter. If passed, the API expects this parameter in the XML format described in Section B.3.1, "Extended Header XML Format". If any other format is used, the API will throw an error and the request will not be submitted.

B.3.1 Extended Header XML Format

```
<extended_header media_type="EMAIL">
    <header_name>email_from_address</header_name>
    <header_value>John.Doe@oracle.com</header_value>
</extended_header>
```

One-to-One Fulfillment Server recognizes the following sets of extended header parameters.

- media_type="EMAIL", "FAX" or "PRINTER"
- email_from_address
- email_reply_to_address

- email_cc_address
- email_bcc_address

If any parameter other than those listed above are passed, the Fulfillment Server will put it into the header information.

Currently the media_type defaults to "EMAIL". If media_type is not set, then the Fulfillment Server will assume it to be EMAIL.

Example B–1 Sample Extended Header XML

```
<extended_header media_type="EMAIL">
    <header_name>email_from_address</header_name>
    <header_value>John.Doe@oracle.com</header_value>
    <header_name>email_cc_address</header_name>
    <header_value>Jane.Doe@oracle.com</header_value>
    <header_name>email_bcc_address</header_name>
    <header_value>xyz@oracle.com</header_value>
</extended_header>
```

B.4 Start_Request

Call the Start_Request first to generate the history request identifier (`x_request_id`) that is used as the tracking identifier for the fulfillment request.

Procedure Specification

```
PROCEDURE Start_Request
( p_api_version IN NUMBER,
p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE,
p_commit IN VARCHAR2 := FND_API.G_FALSE,
p_validation_level IN NUMBER := FND_API.G_VALID_LEVEL_FULL,
x_return_status OUT VARCHAR2,
x_msg_count OUT NUMBER,
x_msg_data OUT VARCHAR2,
x_request_id OUT NUMBER
);
```

Parameter Descriptions

The following table describes the IN parameters associated with the Start Request.

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	Yes	If Default = FND_API.G_FALSEIf is set to true, then the API makes a call to fnd_msg_pub.initialize the message stack. If set to false, then the calling program must initialize the message stack. This action is required to be performed only once, even when more than one API is called.
p_commit	VARCHAR2(1)	No	If Default = FND_API.G_FALSEIf is set to true, then the API commits before returning to the calling program. If set to false, then it is the responsibility of the calling program to commit the transaction.
p_validation_level	NUMBER	No	This value determines whether or not validation should occur. This variable is checked against FND_API.G_VALID_LEVEL_NONE, and if it is greater than the latter, validation occurs.

Parameter Descriptions

The following table lists the OUT parameters associated with Request.

Parameter	Data Type	Description
x_return_status	VARCHAR2	Indicates the return status of the API. The values returned are one of the following:FND_API.G_RET_STS_SUCCESSSuccess: Indicates that the API call was successfulFND_API.G_RET_STS_ERRORExpected Error: There is a validation error, or missing data error.FND_API.G_RET_STS_UNEXP_ERRORUnexpected Error: The calling program can not correct the error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2	Holds the encoded message if x_msg_xount is equal to one.
x_request_id	NUMBER	The fulfillment request identifier. This can be used to track the status of the request.

B.5 Get_Content_Xml

Caution: Oracle does not recommend and does not support the creation of XML fulfillment requests by any other means than through these published APIs. Oracle provides this information on how the XML request is formed solely to facilitate understanding of the fulfillment request process.

Use the Get_Content_XML API to generate one, or more, XML <item> elements for a fulfillment request. The content is formed based on the parameters passed to the API, and the API returns the XML <item> as parameter x_content_xml.

The Get_Content_XML API performs that following tasks:

- Verifies that the p_content_id, p_media_type, p_content_type and p_request_id parameters were passed to it. If not supplied, the API exits with an unexpected error condition.
- Forms the XML element using the parameters passed to it.
- Passes the XML element back to the calling application using the x_ content_xml output parameter.

The Get_Content_XML API generates only one XML parameter for each API call. If you want to submit multiple content items for a single fulfillment request, then you must do the following:

- Append the x_content_xml content from the previous calls to the current one
- Submit the resulting concatenated x_content_xml string.

The Submit_Request API takes this parameter and merges it with the XML request that it forms, submitting the complete request to the Request queue.

Content Source Parameter

New Parameter added in 11.5.9 when One-to-One Fulfillment started supporting both as Marketing Encyclopedia Systems (MES) and Oracle Content Management (OCM).

The Default Value assumed for this parameter is "mes". If GET_CONTENT_XML API is called without this parameter, the repository is assumed to be Marketing Encyclopedia Systems (MES).

From Release 11.5.9 onward, this API supports Oracle Content Management. The other valid value is "ocm".

From 11.5.10 release onward, this API supports the renditions available in Oracle Content Management.

B.6 One-to-One Fulfillment Request Types

Use one of the following content types (p_content_type) to create a valid fulfillment request.

- QUERY
- COLLATERAL
- ATTACHMENT
- DATA

If any other values are passed to it, the Get_Content_XML API exits with an unexpected error.

The following table lists example values for the p_content_id parameters used with query and collateral fulfillment requests.

Example B–2 Sample Parameter Values for Query and Collateral Requests

- p_content_id
- 7008
- 7009
- 7111
- 7234
- 8101
- 9101

B.6.1 Example <item> XML Element

The Get_Content_XML API forms the XML in the fulfillment request.

In Release 11.5.10, Oracle One-to-One Fulfillment supports content from Oracle Content Manager and Management Encyclopedia System. To facilitate this support, the GET_CONTENT_XML has been overloaded. You can use the MES repository without changing any API calls. But, if the new content repository OCM is to be used, then you need to call the overloaded GET_CONTENT_XML with p_content_source = 'ocm' and p_version = version of the document.

The following code is an example of XML that is generated as part of an attachment fulfillment request, with an e-mail media type. The content_id parameter is required for this API. The XML is passed back to the calling application as output parameter x_content_xml.

```
<items><item>
    <media_type>
        <email> abc@oracle.com</email>
    </media_type>
    <item_content quantity="0" user_note="THIS IS A FULFILLMENT TEST"
    source="mes">
        <files>
    <file id="117423" body="yes"/>
    </files>
    </item_content>
</item>
</items>
```

Caution: Oracle does not recommend, nor does it support, the creation of XML fulfillment requests by any other means than through the APIs published in this document. Oracle provides this information on how the XML request is formed *only* to facilitate understanding of the fulfillment request process.

Get_Content_Xml - Overloaded Procedure Specification

```
PROCEDURE Get_Content_XML
( p_api_version IN NUMBER,
p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE,
p_commit IN VARCHAR2 := FND_API.G_FALSE,
p_validation_level IN NUMBER := FND_API.G_VALID_LEVEL_FULL,
x_return_status OUT VARCHAR2,
x_msg_count OUT NUMBER,
x_msg_data OUT VARCHAR2,
p_content_id IN NUMBER,
p_content_nm IN VARCHAR2 := FND_API.G_MISS_CHAR,  -- deprecated
p_document_type IN VARCHAR2 := FND_API.G_MISS_CHAR,--depreciated
p_quantity IN NUMBER := 1,
p_media_type IN VARCHAR2,
p_printer IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_email IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_fax IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_file_path IN VARCHAR2 := FND_API.G_MISS_CHAR,--deprecated
p_user_note IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_content_type IN VARCHAR2,
p_bind_var IN G_VARCHAR_TBL_TYPE := L_VARCHAR_TBL,
p_bind_val IN G_VARCHAR_TBL_TYPE := L_VARCHAR_TBL,
p_bind_var_type IN G_VARCHAR_TBL_TYPE := L_VARCHAR_TBL,
p_request_id IN NUMBER,
x_content_xml OUT VARCHAR2 ,
p_content_source IN VARCHAR2,
p_version IN NUMBER
);
```

Get_Content_Xml Specification

```
PROCEDURE Get_Content_XML
( p_api_version IN NUMBER,
p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE,
p_commit IN VARCHAR2 := FND_API.G_FALSE,
p_validation_level IN NUMBER := FND_API.G_VALID_LEVEL_FULL,
x_return_status OUT VARCHAR2,
x_msg_count OUT NUMBER,
x_msg_data OUT VARCHAR2,
p_content_id IN NUMBER,
p_content_nm IN VARCHAR2 := FND_API.G_MISS_CHAR,  -- deprecated
p_document_type IN VARCHAR2 := FND_API.G_MISS_CHAR,--depreciated
p_quantity IN NUMBER := 1,
p_media_type IN VARCHAR2,
p_printer IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_email IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_fax IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_file_path IN VARCHAR2 := FND_API.G_MISS_CHAR,--deprecated
p_user_note IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_content_type IN VARCHAR2,
p_bind_var IN G_VARCHAR_TBL_TYPE := L_VARCHAR_TBL,
p_bind_val IN G_VARCHAR_TBL_TYPE := L_VARCHAR_TBL,
p_bind_var_type IN G_VARCHAR_TBL_TYPE := L_VARCHAR_TBL,
p_request_id IN NUMBER,
x_content_xml OUT VARCHAR2)
```

Parameter Descriptions

The following table describes the Get_Content_XML IN parameters.

Table B–1 Get_Content_XML IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	Must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number (provided in the documentation).
p_init_msg_list	VARCHAR2	Yes	Default = FND_API.G_FALSE. If set to true, then the API makes a call to fnd_msg_ pub.initialize the message stack. If set to false, then the calling program must initialize the message stack. This action is required to be performed only once, even in cases when more than one API is called.
p_commit	VARCHAR2(1)	No	Default = FND_API.G_FALSE. If set to true, then the API commits before returning to the calling program. If set to false, then it is the responsibility of the calling program to commit the transaction.
p_validation_level	NUMBER	No	This value determines whether or not validation should occur. This variable is checked against FND_API.G_VALID_LEVEL_ NONE, and if it is greater than the latter, then validation occurs.
p_content_id	NUMBER	Yes	A unique numeric identifier that identifies the content that is to be sent out. If not provided in the current API call, then an error message is generated and the API exits.
p_content_nm	VARCHAR2	No	Deprecated, currently unused.
p_document_type	VARCHAR2	No	Deprecated, currently unused.
p_quantity	NUMBER	Yes	Quantity of the content to be sent (set always to 1).

Table B–1 Get_Content_XML IN Parameters (Cont.)

Parameter	Data Type	Required	Descriptions and Validations
p_media_type	VARCHAR2	Yes	Media type of the content generated by the Get_Content_XML API: FAX EMAIL PRINTER. If not provided in the current API call, then an error message is generated and the API exits. The same content can also be sent through all three media types or a combination of two media types. The values can be comma separated, for example, 'FAX,EMAIL,PRINTER' or 'EMAIL,FAX' or 'EMAIL,PRINTER' or any similar combinations.
p_printer	VARCHAR2	No	Supplied printer address, if p_media_ type is PRINTER. This parameter can accept the printer name/alias values.
p_email	VARCHAR2	No	Supplied e-mail address, if p_media_ type is EMAIL.
p_fax	VARCHAR2	No	Supplied fax address, if p_media_ type is FAX
p_file_path	VARCHAR2	No	Deprecated, currently unused.
p_user_note	VARCHAR2	No	Any note that the agent wants to attach.
p_content_type	VARCHAR2	Yes	Valid content types: ▪ QUERY ▪ COLLATERAL ▪ ATTACHMENT ▪ DATA If not provided in the current API call, then an error message is generated and the API exits.
p_bind_var	G_VARCHAR_TBL_ TYPE=L_VARCHAR_ TBL	No	Used with QUERY and DATA fulfillment requests. Part of a set of three matching records.
p_bind_val	G_VARCHAR_TBL_ TYPE=L_VARCHAR_ TBL	No	Used with QUERY and DATA fulfillment requests. Part of a set of three matching records.
p_bind_var_type	G_VARCHAR_TBL_ TYPE=L_VARCHAR_ TBL	No	Used with QUERY and DATA fulfillment requests. Part of a set of three matching records.

Table B–1 Get_Content_XML IN Parameters (Cont.)

Parameter	Data Type	Required	Descriptions and Validations
p_request_id	NUMBER	Yes	The Start_Request API generates this unique identifier for the fulfillment request. If not provided in the current API call, then an error message is generated and the API exits.
p_content_source	VARCHAR2	No	New parameter added in 11.5.9 release. Supports the content repositories MES and OCM. The default is assumed as MES. OCM will use the Rendition functionality in OCM 11.5.10.
p_version	NUMBER	No	OCM repositories support content versioning. Default is assumed as 1.

Get Content XML OUT Parameters

The following table describes the Get_Content_XML OUT parameters.

Parameter	Data Type	Description
x_return_status	VARCHAR2	Indicates the return status of the API. The returned values returned are one of the following: ■ FND_API.G_RET_STS_SUCCESS. Success: Indicates that the API call was successful. ■ FND_API.G_RET_STS_ERROR. Expected Error: There is a validation error, or missing data error. ■ FND_API.G_RET_STS_UNEXP_ERROR. Unexpected Error: The calling program can not correct the error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2	Holds the encoded message if x_msg_count is equal to one.
x_content_xml	VARCHAR2	Output content XML.

Send Request & Submit Request

Send Request is submit request overloaded. Submit Request in turn makes the call to Send Request. The additional parameter is p_preview.

Parameter p_preview

The p_preview parameter provides the ability to preview a fulfillment request before sending it to its destination. You can preview, submit or delete a fulfillment request through the One-to-One Fulfillment Administrator Console.

Previewing allows for a visual inspection of a fulfillment request before you decide to send it or cancel it. When you preview the request in a browser, you can print it through the browser. Printing a previewed request through a browser does not remove the request from the status table.

The value of p_preview determines whether the request is to be dispatched immediately or stored in the database to be inspected visually.

- If p_preview is set to TRUE, then the request is marked as PREVIEW, and written to the Request queue, but it is not sent.
- If p_preview is set to FALSE, then the request is written to the Request queue, and it is sent in the usual fashion.

> **Note:** The fulfillment request is written to the Request queue regardless of the value that is set for the parameter p_preview.

Use the Submit_Request procedure to submit a single new fulfillment request to the fulfillment engine. This API calls the Send_Request API.

Send Request Procedure Specification

```
PROCEDURE Submit_Request
( p_api_version IN NUMBER,
p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE,
p_commit IN VARCHAR2 := FND_API.G_FALSE,
p_validation_level IN NUMBER := FND_API.G_VALID_LEVEL_FULL,
x_return_status OUT VARCHAR2,
x_msg_count OUT NUMBER,
x_msg_data OUT VARCHAR2,
p_template_id IN NUMBER := FND_API.G_MISS_NUM,
p_subject IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_party_id IN NUMBER := FND_API.G_MISS_NUM,
p_party_name IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_user_id IN NUMBER,
p_priority IN NUMBER := G_PRIORITY_REGULAR,
p_source_code_id IN NUMBER := FND_API.G_MISS_NUM,
p_source_code IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_object_type IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_object_id IN NUMBER := FND_API.G_MISS_NUM,
p_order_id IN NUMBER := FND_API.G_MISS_NUM,
p_doc_id IN NUMBER := FND_API.G_MISS_NUM,
p_doc_ref IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_server_id IN NUMBER := FND_API.G_MISS_NUM,
p_queue_response IN VARCHAR2 := FND_API.G_FALSE,
p_extended_header IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_content_xml IN VARCHAR2,
p_request_id IN NUMBER ,
p_preview IN VARCHAR2
);
```

Submit Request Procedure Specification

```
PROCEDURE Submit_Request
( p_api_version IN NUMBER,
p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE,
p_commit IN VARCHAR2 := FND_API.G_FALSE,
p_validation_level IN NUMBER := FND_API.G_VALID_LEVEL_FULL,
x_return_status OUT VARCHAR2,
x_msg_count OUT NUMBER,
x_msg_data OUT VARCHAR2,
p_template_id IN NUMBER := FND_API.G_MISS_NUM,
p_subject IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_party_id IN NUMBER := FND_API.G_MISS_NUM,
p_party_name IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_user_id IN NUMBER,
p_priority IN NUMBER := G_PRIORITY_REGULAR,
p_source_code_id IN NUMBER := FND_API.G_MISS_NUM,
p_source_code IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_object_type IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_object_id IN NUMBER := FND_API.G_MISS_NUM,
p_order_id IN NUMBER := FND_API.G_MISS_NUM,
p_doc_id IN NUMBER := FND_API.G_MISS_NUM,
p_doc_ref IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_server_id IN NUMBER := FND_API.G_MISS_NUM,
p_queue_response IN VARCHAR2 := FND_API.G_FALSE,
p_extended_header IN VARCHAR2 := FND_API.G_MISS_CHAR,
p_content_xml IN VARCHAR2,
p_request_id IN NUMBER
);
```

Send and Submit_Request IN Parameters

The following table lists the IN parameters for Send_Request and Submit_Request.

Table B–2 Send and Submit_Request IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number (provided in the documentation)
p_init_msg_list	VARCHAR2	Yes	Default = FND_API.G_FALSE. If set to true, then the API makes a call to fnd_msg_pub.initialize the message stack. If set to false, then the calling program must initialize the message stack. This action is required to be performed only once, even in the case where more than one API is called
p_commit	VARCHAR2(1)	No	Default = FND_API.G_FALSE. If set to true, then the API commits before returning to the calling program. If set to false, then it is the calling program's responsibility to commit the transaction.
p_validation_level	NUMBER	No	This value determines whether validation should occur or not. This variable is checked against FND_API.G_VALID_LEVEL_NONE and if it is greater than the latter, validation occurs.
p_template_id	NUMBER	No	The fulfillment request template identifier. NULL if request does not correspond to a predefined template.
p_subject	VARCHAR2	No	Subject specifies the purpose of the request
p_party_id	NUMBER	No	Customer Identifier.
p_party_name	VARCHAR2	No	Customer Name.
p_user_id	NUMBER	Yes	Agent or user Identifier who is making the request.
p_priority	NUMBER	No	Priority for that particular request. If none provided, default is regular

Table B–2 Send and Submit_Request IN Parameters (Cont.)

Parameter	Data Type	Required	Descriptions and Validations
p_source_code_id	NUMBER	No	Campaign or promotion field.
p_source_code	VARCHAR2	No	Campaign or promotion field.
p_object_type	VARCHAR2	No	Campaign or promotion field.
p_object_id	NUMBER	No	Campaign or promotion field.
p_order_id	NUMBER	No	Unique identifier of the field.
p_doc_id	NUMBER	No	Document Identifier, if not provided, set to p_request_id
p_doc_ref	VARCHAR2	No	Document Reference. If not provided, set to JFUF.
p_server_id	NUMBER	No	Unique Identifier of the One-to-One Fulfillment Server. It is the primary key of JTF_FM_SERVICE_ALL table. If this is provided, that particular server processes the request.
p_queue_response	VARCHAR2	No	Deprecated and not supported. This field is used internally for other purposes.
p_extended_ header	VARCHAR2	No	Used to insert a character string in the form of one or more extended header elements into the XML request.
p_content_xml	VARCHAR2	Yes	The XML content formed by calling the GET_CONTENT_XML API.
p_request_id	NUMBER	Yes	The unique identifier for the fulfillment request generated by the Start_Request API. If not provided in the current API call, an error message is generated and the API exits.
p_preview	VARCHAR2	Yes	Boolean Value: Used to specify whether or not this is a preview request: TRUE: Preview this document. FALSE: Do not preview this document, instead send it to the fulfillment request queue.

Submit Request OUT Parameters

The following table describes the Submit_Request OUT parameters.

Parameter	Data Type	Description
x_return_status	VARCHAR2	Indicates the return status of the API. Returns one of the following values: ■ FND_API.G_RET_STS_SUCCESS. Success: Indicates that the API call was successful ■ FND_API.G_RET_STS_ERROR. Expected Error: There is a validation error, or missing data error. ■ FND_API.G_RET_STS_UNEXP_ERROR. Unexpected Error: The calling program cannot correct the error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2	Holds the encoded message if x_msg_count is equal to one.

B.7 Messages and Notifications

The APIs contained in package JTF_FM_REQUEST_GRP generate messages and notifications as necessary. In addition, some messages are common to all of the Fulfillment APIs. These messages are detailed in the following sections.

The messages and notifications generated by each API are described in the following sections.

- Section B.7.1, "Common Messages"
- Section B.7.2, "Get_Content_XML Messages"

Note: Not all status notifications are required to provide a number identifier. However, in many cases a number identifier is provided.

B.7.1 Common Messages

The following table lists the messages and notifications that are common to many of the APIs contained in package JTF_FM_REQUEST_GRP.

Table B–3 Common Status Messages

NUMBER	TYPE	NAME	TEXT
210925	E	JTF_FM_API_QUEUE_NOT_FOUND	Could not find queue names for Server ID &ARG1
210927	E	JTF_FM_API_REQUEST_ NOTFOUND	Could not find fulfillment request for Request ID & ARG1 in database
210932	E	JTF_FM_API_MISSING_REQUEST_ID	Value for Request ID cannot be NULL.
210939	E	JTF_FM_API_DEBUG_MESSSAGE	1 to 1 Fulfillment API: &ARG1
210940	S	JTF_FM_API_SUCCESS_MESSAGE	The &ARG1 API has completed successfully.

B.7.2 Get_Content_XML Messages

The following table lists the messages and notifications that are used by the Get_Content_XML API.

Table B–4 Status Messages for Get_Content_XML

NUMBER	TYPE	NAME	TEXT
210922	E	JTF_FM_API_INVALID_MEDIATYPE	Invalid media type &ARG1 passed for Content ID &ARG2. Allowed media types are EMAIL, FAX, PRINTER
210923	E	JTF_FM_API_MULTIPLE_MEDIATYPE	Multiple media types were passed for Content ID = &ARG1. Only one media type is allowed for each content.
210924	E	JTF_FM_API_INVALID_CONTENT	Invalid content type &ARG1 passed for Content ID &ARG2. Allowed content types are QUERY, COLLATERAL, ATTACHMENT or DATA.

Table B–4 Status Messages for Get_Content_XML (Cont.)

NUMBER	TYPE	NAME	TEXT
210929	E	JTF_FM_API_MISSING_CONTENT_ID	Value for Content ID cannot be NULL.
210930	E	JTF_FM_API_MISSING_MEDIA_TYPE	Value for Media Type cannot be NULL.
210931	E	JTF_FM_API_MISSING_CONTENTTYPE	Value for Content Type cannot be NULL.
210974	E	JTF_FM_DISP_CONTENT_NOT_FOUND	Cannot find document for Content ID &ARG1.
	E	JTF_FM_API_CONTENT_INVALID	The content_id &ARG1 is not a valid content_ id, or does not exist.
	E	JTF_FM_API_INVALID_LISTTYPE	Invalid list type &ARG1 passed. Allowed list types are VIEW or ADDRESS
	E	JTF_FM_API_MISSING_LIST_TYPE	Value for List cannot be NULL
	E	JTF_FM_API_INVALID_REQTYPE	Invalid request type specified. Only allowed values are "E,P or T". E for "Electronic Request", P for "Physical Collateral" and T for "Test Requests".
	E	JTF_FM_API_CF_INVALID_MEDIA	Invalid Media Type specified. Only allowed Media types are any one of the following. "EPF, EFP, FEP, FPE, PEF, PFE, EP, EF, E, PE, PF, P, FE, FP, F". E represents "EMAIL", P represents "PRINTER", and F represents "FAX"

B.8 Example Scripts

The following scripts are examples that may not work if used in the exact form shown here and placed into another environment.

Sample Script - Start_Request, Get_Content_Xml, Submit_Request

```
DECLARE
l_api_versionNUMBER := 1.0;
l_commitVARCHAR2(5) := FND_API.G_TRUE;
l_content_idVARCHAR2(30);
l_media_typeVARCHAR2(30);
l_request_typeVARCHAR2(20);
l_user_noteVARCHAR2(1000);
l_document_typeVARCHAR2(150);
l_bind_var    JTF_FM_REQUEST_GRP.G_VARCHAR_TBL_TYPE ;
l_bind_var_type    JTF_FM_REQUEST_GRP.G_VARCHAR_TBL_TYPE ;
l_bind_val    JTF_FM_REQUEST_GRP.G_VARCHAR_TBL_TYPE ;
l_template_id  NUMBER;
l_party_id NUMBER;
l_user_idNUMBER;
l_server_id NUMBER;
l_request_idNUMBER;
l_subjectVARCHAR2(100);
l_msg_count  NUMBER;
l_msg_data VARCHAR2(1000);
l_return_status VARCHAR2(1000);
l_content_xml1 VARCHAR2(1000);
l_content_xml VARCHAR2(10000);
l_content_nmVARCHAR2(100);
l_emailVARCHAR2(100);
l_printerVARCHAR2(100);
l_file_pathVARCHAR2(100);
l_faxVARCHAR2(100);

BEGIN
    -- Start the fulfillment request. The output request_id must be passed
    -- to all subsequent calls made for this request.
JTF_FM_REQUEST_GRP.STart_Request
(
    p_api_version => l_api_version,
    x_return_status => l_return_status,
    x_msg_count => l_msg_count,
    x_msg_data => l_msg_data,
    x_request_id => l_request_id
);

DBMS_OUTPUT.PUT_LINE('Request_ID: '||to_char(l_request_id));
-- Initialize the parameters for Master Document.
              -- this does not need anything.it just sends the email .
```

```
   l_content_id := &p_contentid;--item id
    l_media_type := &p_mediaType;
   l_document_type := &docType;--zip or htm or html
   l_request_type := &p_requestType;
   l_user_note := 'Please dont hesitate to call at 1-800-INFODSK.';
   l_email := &p_email;
   l_printer  := '\\ORADEV8\DEVSTANLEY';
  l_file_path := NULL;
   l_fax  := '9999999999';

    -- This call gets the XML string for the content(Master Document) with
    -- the parameters as defined above
JTF_FM_REQUEST_GRP.Get_Content_XML
(
    p_api_version => l_api_version,
    x_return_status => l_return_status,
    x_msg_count => l_msg_count,
    x_msg_data => l_msg_data,
    p_content_id => l_content_id,
    p_content_nm => l_content_nm,
    p_document_type =>  l_document_type,
    p_media_type=> l_media_type,
    p_printer => l_printer,
    p_email => l_email,
    p_file_path => l_file_path,
    p_fax => l_fax,
    p_user_note => l_user_note, p_content_type => l_request_type,
    p_bind_var => l_bind_var,
    p_bind_val => l_bind_val,
    p_bind_var_type => l_bind_var_type,
    p_request_id => l_request_id,
    x_content_xml => l_content_xml1);

-- The output XML string is stored in a local variable
                      l_content_xml := l_content_xml1;

-- Check the return status
DBMS_OUTPUT.PUT_LINE('Return Status: '||l_return_status);
DBMS_OUTPUT.PUT_LINE('Message_Count: '||l_msg_count);
DBMS_OUTPUT.PUT_LINE('Message Data: '||l_msg_data);
 -- Initialize Parameters for submitting the fulfillment request
l_user_id := &p_userId;--this is a parameter that can be filled during run
-- l_server_id := null;
l_subject := &p_subject;--subject could be environment and what test it is
```

```
-- Submit the fulfillment request
JTF_FM_REQUEST_GRP.Submit_Request
(
    p_api_version => l_api_version,
    p_commit => l_commit,
    x_return_status => l_return_status,
    x_msg_count => l_msg_count,
    x_msg_data => l_msg_data,
    p_subject => l_subject,
    p_party_id => l_party_id,
    p_user_id => l_user_id,
    p_queue_response => FND_API.G_TRUE,
    p_content_xml => l_content_xml,
    p_request_id => l_request_id
);

DBMS_OUTPUT.PUT_LINE('Return Status: '||l_return_status);
DBMS_OUTPUT.PUT_LINE('Message_Count: '||l_msg_count);
DBMS_OUTPUT.PUT_LINE('Message Data: '||l_msg_data);
DBMS_OUTPUT.PUT_LINE('Message Data: '||l_user_id);

END;
```

Glossary

Advanced Queue

A mechanism within the Oracle database that is used by the Fulfillment APIs and Fulfillment Server to queue requests. Fulfillment APIs put requests into the queue. The Fulfillment Server pulls requests out of the queue.

bounce back

The process of returning e-mail to the sending server because of an incorrect address (hard bounce back) or because the recipient's inbox is full (soft bounce back).

content item (cover letter)

An instance of a content type.

content type

The definition of the structure of a cover letter that can be used in e-mail, fax and print.

database connectivity file

A text file that contains the connection parameters for the Oracle Applications database. It is created automatically when Oracle Applications is installed.

data source file

A text file that contains tab-separated merge fields that have been obtained by a query.

dequeuing

A process by which the Fulfillment Server retrieves fulfillment requests from a queue.

enqueuing

A process by which a business application places fulfillment requests into a queue.

e-mail channel

Requests sent by e-mail.

fax channel

Requests sent by fax.

Fulfillment Administration Console

The HTML interface for administrators to define and maintain the servers and the associated e-mail, fax and print servers. Administrator can define groups of agents and associate them with the defined servers. Oracle One-to-One Fulfillment supports two content repositories, Marketing Encyclopedia System (MES) and Oracle Content Manager (OCM), and provides administration HTML pages to create and upload content into each repository.

fulfillment

The process of compiling and distributing information to customers.

fulfillment request

The customer request that is processed. The request can be made up of a simple e-mail, merged e-mail, e-mail with attachments, and so on. A request starts when the Fulfillment APIs receive data. The request ends when the Fulfillment Server passes information to the Java mail APIs or to the Print or Fax Enabler.

Fulfillment Server

The multi-threaded Java server that processes Fulfillment requests by de-queuing them from an Advanced Queue.

glue code

Code that connects two components.

Java

A technology that is both a programming language and a platform.

Java API

The Java application programming interface. A large collection of ready-made software components that provides many capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces known as packages.

javac

The Java programming language compiler.

Java language

An object programming language in which programs are both compiled and interpreted. Using the compiler, a program is translated into an intermediate language called Java bytecodes, which are platform-independent codes that are interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java bytecode instruction on the computer. Compilation happens only once, but interpretation occurs each time the program runs.

Java platform

A software-only environment that runs on top of other hardware-based platforms. The Java platform has two components: Java Virtual Machine (Java VM) and the Java Application Programming Interface (Java API).

Java Virtual Machine

The base for the Java platform. It is ported onto various hardware-based platforms.

malformed e-mail

E-mail addresses that are incorrectly formed (such as a missing "@"). The resulting e-mail cannot be sent to the customer.

mass request type

Large volume One-to-One requests. For mass requests, the calling application passes a SQL query which when executed derives the list of e-mail addresses and fax numbers.

master document

An HTML file that contains the main content in a fulfillment item.

Oracle Marketing Encyclopedia System

A file repository that stores master documents and queries that are used to specify fulfillment content.

multi channel

Requests sent by more than one channel.

OCM

See Oracle Content Manager.

(One-to-One) request

Requests initiated by a calling application (such as Oracle Collections) to send one ore more master documents or pieces of collateral to a customer.

(One-to-One) job

Requests that are comprised of one or more jobs (sub-requests) which represent the individual e-mail or faxes, and so forth, that are sent to a specific customer.

opened e-mail

E-mails sent by Oracle One-to-One Fulfillment that have been opened by the recipient.

Oracle Content Manager

A document repository for fulfillment content.

physical request

One-to-One physical collateral requests for Collateral kits. These Collateral kits are separated into individual collateral and an order created which has multiple order lines (one for each collateral) and multiple shipment lines if the request has multiple recipients.

print channel

Requests sent to a printer and then mailed to a customer.

query

A SQL (structured query language) statement that generates selected data from a database. A query determines which records meet the defined criteria, and which fields of those records are returned.

queue

A repository for messages. Queues are stored in queue tables. Each queue table is a database table and contains one or more queues.

sent e-mail

E-mails sent by Oracle One-to-One Fulfillment.

single request type

One-to-One requests sent to a single customer.

template

A grouping of material that is useful but not required for distributing master documents and collateral. An Oracle Marketing Encyclopedia System template is a package of content that is compiled by the Fulfillment Server. The fulfillment request identifies the template being used to compile the content.

unopened e-mail

E-mails sent by Oracle One-to-One Fulfillment that have not yet been opened by the recipient.

Unsubscribe option

An option for e-mail sent by Oracle One-to-One Fulfillment that contain an "Unsubscribe" link, and in which the recipient (customer) has selected to not be contacted, either by e-mail or not at all.

versioning

Updating a content item by modifying an existing version or creating a new one. Each version can have the status of In Progress, Submitted or Approved. In Oracle Content Manager, multiple approved versions can exist, but only one version can be set as the live version.

Index