Oracle9*i*AS Syndication Server

User's and Administrator's Guide

Release 2 (9.0.2)

January 2002

Part No. A95917-01

As a feature of Oracle9*i*AS, Oracle9*i*AS Syndication Server is designed to deliver file system and database content to Information and Content Exchange (ICE)-compliant subscriber applications, and automatically provides content updates over networks supporting the HTTP or HTTPS protocol in a secure way. It provides extensibility at multiple levels; it supports HTTP, HTTPS, and SMTP communication mechanisms with the content subscriber, including the Information and Content Exchange (ICE) Version 1.1 protocol. It allows access using corporate databases and conventional file systems. Oracle9*i*AS Syndication Server features a comprehensive administration system to persistently maintain subscriptions and profiles for subscribers and content providers. Subscriber applications can choose to actively request content from Oracle9*i*AS Syndication Server or have it automatically delivered to them.

ORACLE

Oracle9*i*AS Syndication Server User's and Administrator's Guide, Release 2 (9.0.2)

Part No. A95917-01

Primary Author: Alok Srivastava, Cheng Han, Xiaohua Lu, Rod Ward

Contributing Authors: Timothy Chien, Michael Sekurski, Joseph Meeks, Larry Guros, Susan Shepard

Contents

List of Figures

List of Tables

Send Us Your Comments

Oracle9*i*AS Syndication Server User's and Administrator's Guide, Release 2 (9.0.2)

Part No. A95917-01

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

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

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

- Electronic mail: nedc-doc_us@oracle.com
- FAX: 603.897.3825 Attn: Oracle9*i*AS Syndication Server Documentation
- Postal service:
  Oracle Corporation
  Oracle9*i*AS Syndication Server Documentation
  One Oracle Drive
  Nashua, NH 03062-2804
  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

As a feature of Oracle9*i*AS, Oracle9*i*AS Syndication Server is designed to deliver file system and database content to Information and Content Exchange (ICE)-compliant subscriber applications, and automatically provides content updates over networks supporting the HTTP or HTTPS protocol in a secure way. It provides extensibility at multiple levels; it supports HTTP, HTTPS, and SMTP communication mechanisms with the content subscriber, including the Information and Content Exchange (ICE) Version 1.1 protocol. It allows access using corporate databases, and conventional file systems. Oracle9*i*AS Syndication Server features a comprehensive administration system to persistently maintain subscriptions and profiles for subscribers and content providers. Subscriber applications can choose to actively request content from Oracle9*i*AS Syndication Server, or have it automatically delivered to them.

Intended Audience

This guide is for developers who want to easily and more quickly develop and manage content subscription services for subscribers accessing information from content providers. An understanding of Oracle9*i*, Oracle9*i*AS, Java, and XML is required.

This guide is also for administrators who will be managing the Oracle9*i*AS Syndication Server system using the Syndication Server Manager, a Web-based tool available through Oracle Enterprise Manager (OEM).

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our

documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

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

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

Organization

This guide contains the following chapters and appendixes:

Chapter 1	Introduces Oracle9*i*AS Syndication Server; explains concepts.
Chapter 2	Describes how to configure Oracle9*i*AS Syndication Server.
Chapter 3	Describes the development flow for setting up a content syndication system.
Chapter 4	Describes Oracle9*i*AS Syndication Server administration tasks.
Appendix A	Describes the ICE subscriber application development kit.
Appendix B	Describes some frequently asked questions (FAQ).
Appendix C	Describes some helpful links to W3C and ICE specifications.
Appendix D	Describes Oracle9*i*AS Syndication Server error messages.
Appendix E	Describes extensible features.
Appendix F	Describes how to manually deploy Oracle9*i*AS Syndication Server.

Glossary	Describes content syndication and Oracle9*i*AS Syndication Server terms.

Related Documentation

> **Note:** See your operating-system specific installation guide for more information.
>
> For the latest documentation, see the Oracle Technology Network Web site:
>
> `http://otn.oracle.com/`

For more information, see the following manuals:

- *Oracle9i XML Reference*
- *PL/SQL User's Guide and Reference*
- *Oracle9i Java Developer's Guide*
- *Oracle9i Java Stored Procedures Developer's Guide*
- *Oracle9i Enterprise JavaBeans Developer's Guide and Reference*
- *Oracle9i JDBC Developer's Guide and Reference*
- *Oracle9i SQLJ Developer's Guide and Reference*

Conventions

The following conventions are used in this guide:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted

Convention	Meaning
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
< >	Angle brackets enclose user-supplied names.
[ ]	Brackets enclose optional clauses from which you can choose one or none.

1
Introduction

As a feature of Oracle9*i*AS, Oracle9*i*AS Syndication Server is designed to deliver file system and database content to Information and Content Exchange (ICE)-compliant subscriber applications, and automatically provides content updates over networks supporting the HTTP or HTTPS protocol in a secure way. It provides extensibility at multiple levels; it supports HTTP, HTTPS, and SMTP communication mechanisms including the ICE Version 1.1 protocol. It allows access using corporate databases and conventional file systems. Oracle9*i*AS Syndication Server features a comprehensive system for the automated, controlled exchange, and management of digital assests among business partners.

1.1 Content Syndication Concepts

Content syndication is the aggregation, exchange, and distribution of information from content providers to subscriber applications. The content providers provide the content, the syndicators distribute the content, and the subscriber applications, who subscribe to a set of content offerings from a catalog, reuse the content.

With the advent of the Internet, syndication is rapidly evolving to:

- Automate the process of information aggregation and exchange
- Handle a wide variety of content formats
- Deliver content to a wide variety of distribution channels automatically, securely, and reliably

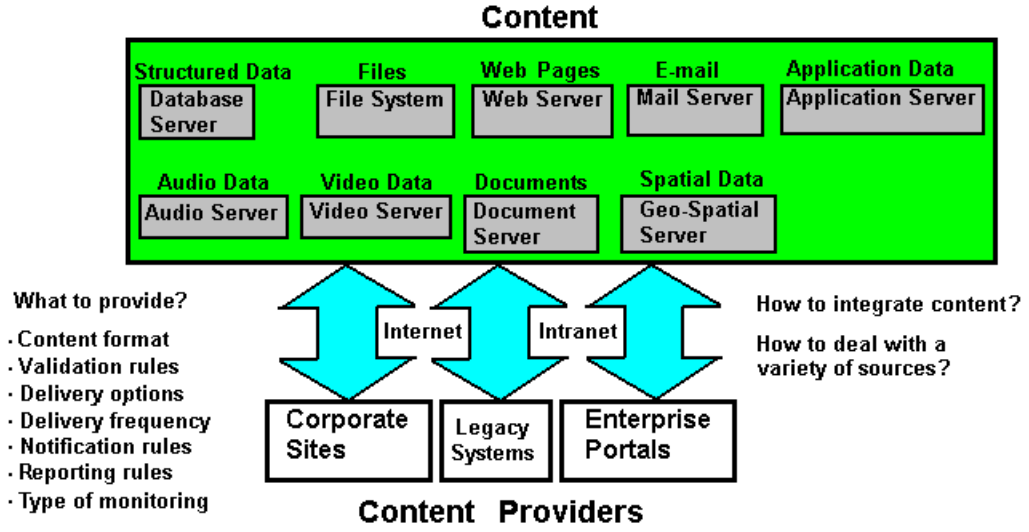
A content subscriber application acquires a content catalog of available offers from a content syndicator and selects the desired subscription offers. After establishing all contractural, monetary, and business agreements between the subscriber application and syndicator, the subscription process is complete. As part of this subscription, a

subscriber application may wish to pull new information or have it automatically provided to them either when it is updated, or at some specified time interval.

1.1.1 Business Problems or Technical Challenge

Typically, sharing content (information) among affiliated groups of **content providers** through **affiliated networks** is an expensive, impromptu process. Each new partner requires customized, manual, time-consuming processes to update, share, and exchange content. As Figure 1–1 shows, the content syndicator struggles with each new affiliate member over content format, validation, delivery options and frequency, notification, reporting and monitoring, content integration into disparate Web servers, operating systems, databases, and other proprietary technology. Content comes from a variety of sources including dialup and ISDN connections, satellite dishes, FM broadcasts, SMTP e-mail servers, FTP servers, hard disks, and magnetic tapes.

Figure 1–1 Typical Problems Facing Content Syndicators



A comprehensive syndication framework must not only address the issues of disparate content formats and communication channels, but also the needs for reliability, scalability, and security in an enterprise environment.

1.1.2 Standardized Solution Emerges

The Information and Content Exchange (ICE) standard (Version 1.1) manages and automates the establishment of syndication relationships, content transfer, and results analysis. The purpose of ICE is to replace the impromptu efforts of content exchange with a standardized, low-cost mechanism for managed, automated content exchange and content sharing of Web site assets among partners. Through the adoption of an industry-specific vocabulary, ICE provides a complete solution for syndicating any type of information between content providers and subscriber applications. Subscriber applications include networked partners and their affiliates.

As ICE becomes universally accepted and implemented across the Web, it will enable companies and industries of all sizes to take advantage of the vast amount of content on the Web, and establish **business-to-business value chains** in a low-cost, automated way. Web application developers can use ICE as a standard platform to exchange multiple data types and rapidly deploy applications, while protecting data privacy and incorporating existing standards. ICE dramatically reduces the cost and difficulty of creating and operating online distribution networks, and building value chains among content providers, affiliated networks, syndicators, and subscriber applications. ICE increases the value of business alliances by facilitating the controlled exchange and management of content among affiliated networks. Businesses can form partnerships with multiple affiliates at minimal incremental cost.

ICE defines a complete server-to-server syndication protocol and processing model. Within ICE, an XML-based common language and architecture is used to describe groups of content offerings as catalogs, as well as to schedule content delivery (*push* and *pull*), and to update type (incremental versus full), business rules, intellectual property rights, and all other aspects of automated content exchange.

ICE uses XML document exchange as its fundamental protocol model. ICE messages, also known as **payloads**, are valid XML documents, with a single ICE-payload root element and a structured hierarchy of tags describing the ICE operations and data. A payload is a single instance of an XML document formatted according to the protocol definitions contained in the ICE specification.

Payloads are transported over HTTP and use a sequenced package model. The two basic ICE actions are ***push*** and ***pull***. To send an ICE / HTTP payload, the sender performs an HTTP POST to a URL provided by the receiver.

ICE requests are specified using an ICE-request XML element, and ICE responses are specified using an ICE-response element. For ICE / HTTP, the ICE-request must be sent in an HTTP POST, and the ICE-response to that request must be sent in the

HTTP response to that POST. Therefore, a single ICE request/response pair always maps directly to a single HTTP POST/corresponding response pair. Event logs are exchanged automatically for helping exception handling or problem diagnostics.

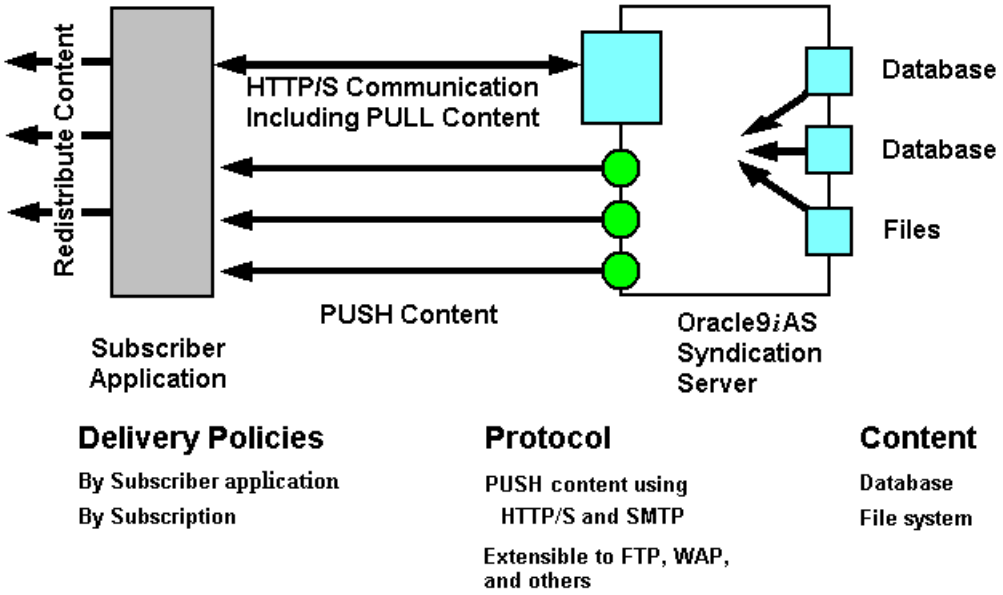
1.2 Syndication Server Solution

Oracle9*i*AS Syndication Server is designed to deliver file system and database content to ICE-compliant subscriber applications and automatically provides content updates over networks supporting the HTTP or HTTPS protocol. This radically simplifies the process of syndication. Oracle9*i*AS Syndication Server provides a comprehensive solution for content aggregation, syndication, and distribution by letting you make available any or all of your content, to anywhere, at anytime, and deliver it securely. Content syndicators can use Oracle9*i*AS Syndication Server with the following benefits:

- Aggregate content from various content repositories, including existing databases, enterprise applications, and e-mail repositories. Using Syndication Server to extract or adapt content from database and file system sources, this content is consolidated into a single point of access known as a content catalog offering, accessible to any ICE-compliant content subscriber application.
- Deliver content according to policies associated with each subscription, and conforming to the ICE protocol. Content can be delivered according to a prescribed schedule or upon a *pull* request from the subscriber application.
- Administer all server functions from Oracle Enterprise Manager, including subscriber application and content provider registration, subscription delivery schedule and status, and system event monitoring.

Figure 1–2 shows the key features of Oracle9*i*AS Syndication Server.

Figure 1–2 Key Features of Oracle9iAS Syndication Server



1.3 Overview of Concepts

Oracle9*i*AS Syndication Server consists of the following components:

- Transport protocol manager
- Request manager
- Subscription manager
- Affiliates manager
- Message manager
- Content provider adaptor

Table 1–1 describes each of these components.

Table 1–1 Syndication Server Components and Their Functions

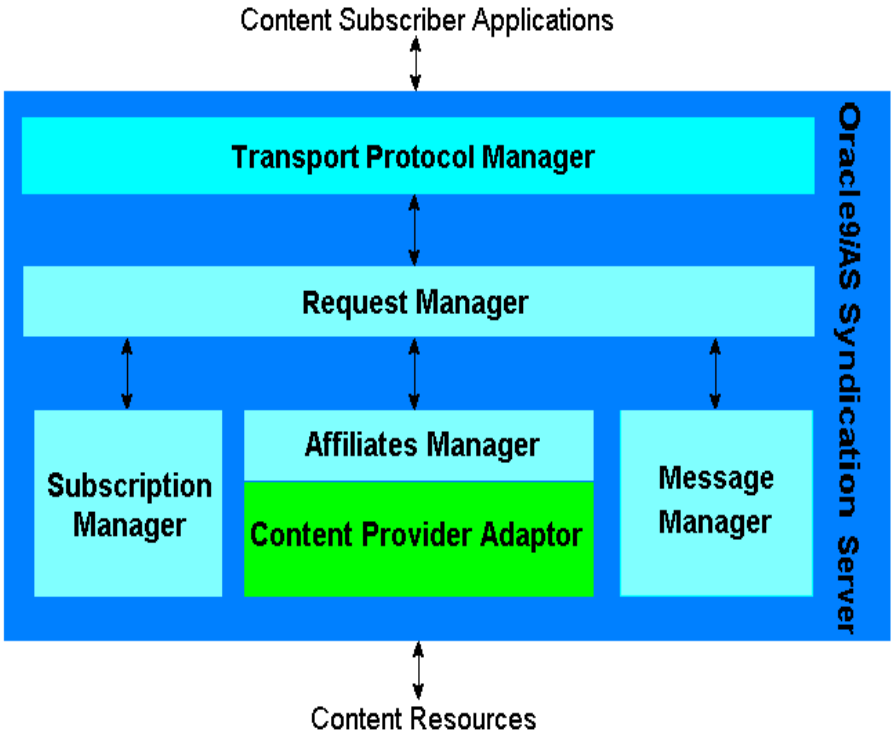
Components	Functions
Transport protocol manager	Handles both the *pull* and *push* content deliveries from Syndication Server to subscriber applications and transports the content over a specified transport layer, such as HTTP or HTTPS.
Request manager	Coordinates the handling of request messages from content subscriber applications and response messages from content providers through the transport protocol manager. Interacts with the transport protocol and affiliates managers. Serves as a central dispatcher of messages.
Subscription manager	Manages the subscriptions of each subscriber application by maintaining mappings between subscriptions and catalog offers.
Affiliates manager	Manages all the profiles of the content providers. Each content provider profile includes its preferred content provider adaptor implementation and all necessary parameters to this adaptor, its delivery policy, its business terms, and its account and billing preferences.
Message manager	Manages protocol-specific messages, such as ICE messages, and automatic packaging of digital content.
Content provider adaptor	Provides the interface for the Syndication Server engine to access each registered content provider as a set of content provider operation modules. The content provider adaptor supports both the file and database types of content through file and database content provider operation modules.

Figure 1–3 shows the Oracle9*i*AS Syndication Server architecture.

Oracle9*i*AS Syndication Server is a fully J2EE-compliant servlet deployed in Oracle9*i*AS Containers for Java (OC4J) as part of the Oracle9*i*AS Portal and Wireless installation, and can take advantage of OC4J services for session and failover management. Syndication Server can also be deployed as needed both within OC4J and across multiple Oracle9*i*AS instances to accommodate varying machine load and utilization from syndication processes. To maintain subscriber application credentials for accessing specific catalog offerings from content providers, Syndication Server uses user authentication and authorization features of the Oracle9*i*AS repository.

All information about subscriber applications, their subscriptions, and content resources are stored in the Syndication Server registry residing in the Oracle9*i*AS repository.

Figure 1–3 Syndication Server Architecture



1.4 Roles in Content Exchange Scenario

Section 1.4.1 and Section 1.4.2 describe the roles of content subscriber applications and content provider adaptors.

1.4.1 Content Subscriber Applications

In the business of content syndication, a content subscriber application is one of the two parties who obtains and repackages information and content from a content

syndicator. A subscriber application can be an ICE-compliant application or a non ICE-compliant application that uses the client library API (see Appendix A).

1.4.2 Content Provider Adaptor

Content providers provide the content. Syndication Server provides two types of content providers: file and database. Each type of content provider has the following minimum set of content provider operation modules:

- Catalog module

  The Catalog module is used for content providers to provide catalog information of their available subscription offers. This module is used to enforce the uniform input and output formats for all individually developed catalog modules. However, each content provider defines its own semantics of a subscription offer and offer group.

- Content Access module

  The Content Access module takes the unique subscription ID as input in order to collect all the related information, and constructs a response ICE payload according to the given module interface. Subscriber applications can initiate the process of *pulling* content from Syndication Server, or have content *pushed* to them. Either path of content access executes a Content Access module.

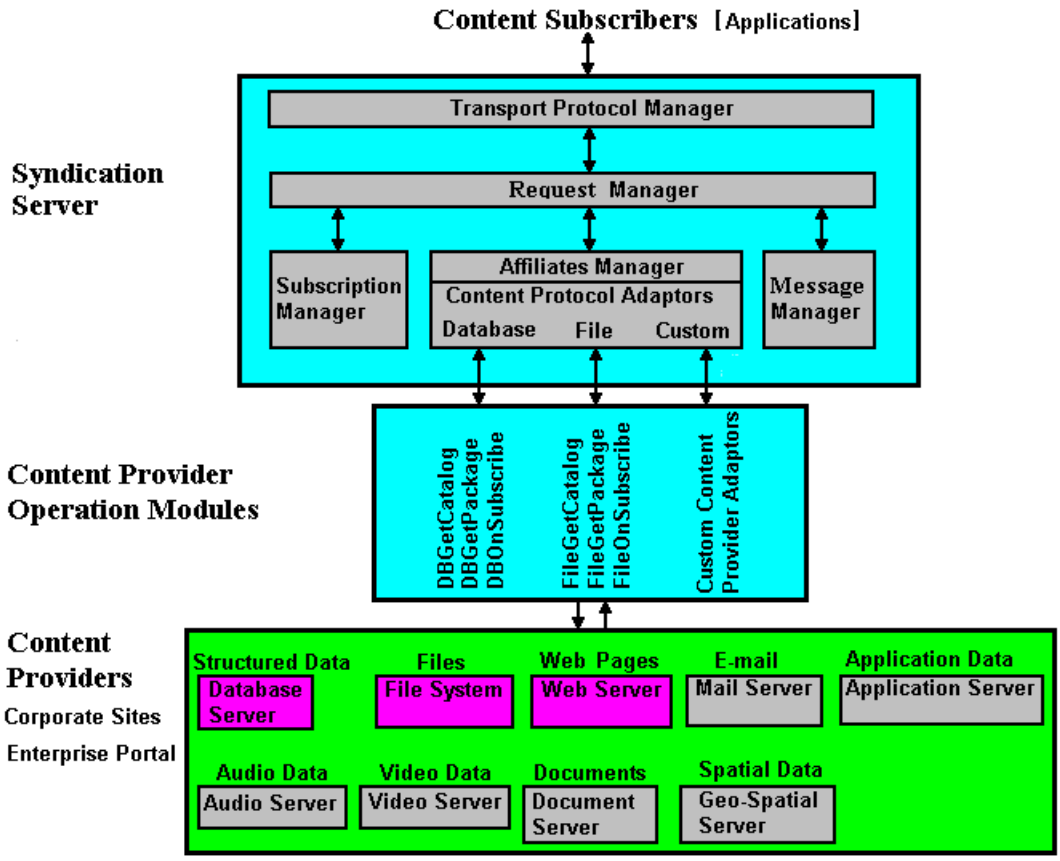
Figure 1–4 shows the architecture of Oracle9*i*AS Syndication Server and how the content provider adaptors relate to specific content provider operation modules designed for Syndication Server.

1.4.3 Syndicator

Syndicators automate the process of syndication. The role of the syndicator is to:

- Deploy the Oracle9*i*AS Syndication Server runtime system
- Distribute content from content providers to subscriber applications
- Monitor Syndication Server performance and log activity

Figure 1–4 Oracle9iAS Syndication Server Architecture



1.5 Syndication Server Operations Overview

Section 1.5.1through Section 1.5.5 describe and explain how the major operations of Syndication Server work, beginning with processing a request from a subscriber application who wants to access a specific content resource using Syndication Server. The operations involve:

- Setting up a subscriber application account on Syndication Server
- Aggregating subscription offerings and generating the catalog

- Process subscription requests
- Accessing content for that subscription

1.5.1 Setting Up a Subscriber Application Account

Once the subscriber application and syndicator have worked out all contractual, monetary, and business implications, then a subscriber application account can be created to allow the subscriber application to access Syndication Server. The Syndication Server administrator creates a user account for the subscriber application and sends back a confirmation message. This confirmation message contains information about the subscriber application's account, such as his subscriber application ID, contact information, access control information, and how to obtain that catalog.

1.5.2 Aggregating Subscription Offerings and Generating the Catalog

With his new subscriber account ID, the subscriber application may make a get-catalog request to view all the subscription offerings. Syndication Server contacts all authorized content providers for available offerings. After receiving all responses from the content providers, Syndication Server constructs a single catalog response and returns it to the subscriber application. Each content catalog is considered a single offer group in the generated catalog response, which is marked by that content provider's unique ID. Syndication Server can aggregate content catalogs from a variety of content providers.

1.5.3 Process Subscription Requests

The subscriber application reviews the catalog and chooses one single content offer group. The subscriber application then supplies additional information as to a preference for negotiable parameters such as delivery policy and business terms, and then makes a request to get approval for a subscription. Syndication Server (the syndicator) invokes the subscription manager to process the request, and possibly involves the corresponding content provider as needed. Following a possible negotiation process with the content provider, a subscription agreement is made, and having agreed to a set of terms, the subscriber application receives a message indicating that the subscription is set and active.

1.5.4 Accessing Content for a Subscription

If the subscription is enabled for *pulling* content delivery, the subscriber application initiates a content access request and provides the subscription ID. Syndication

Server *pulls* content from the corresponding content provider and returns it to the subscriber application.

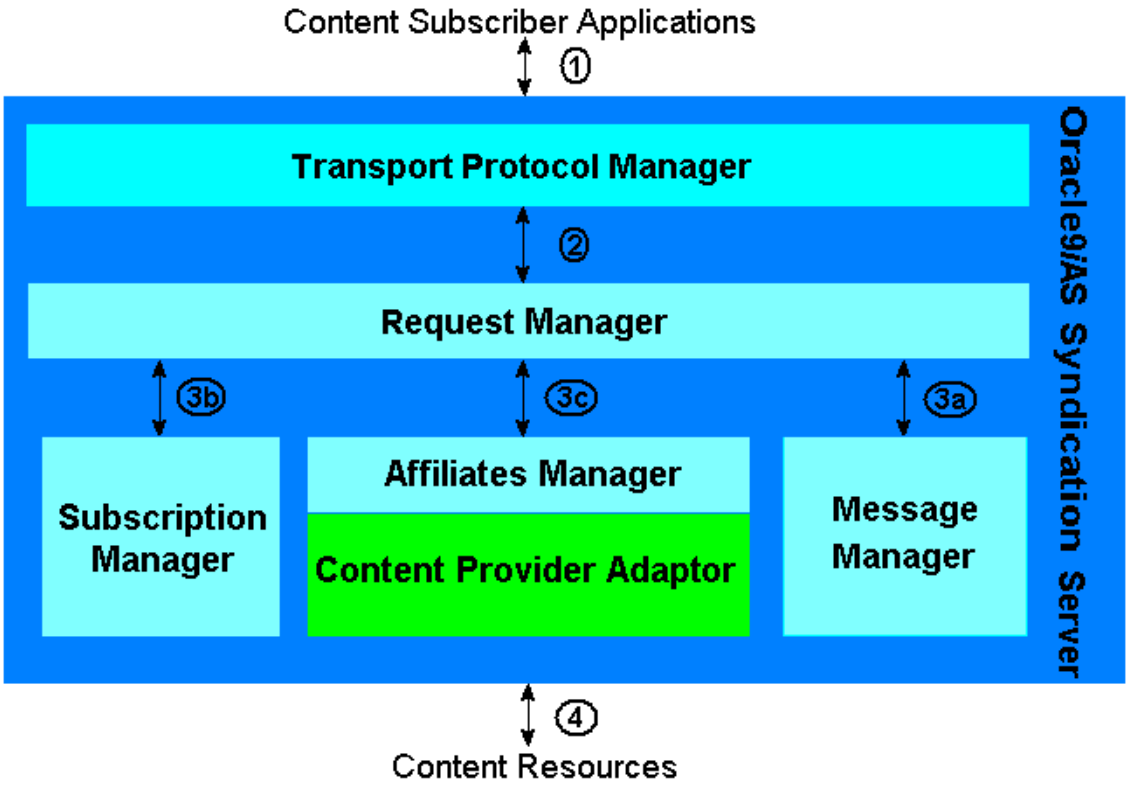
If the subscription is enabled for automatically *pushing* content to the subscriber application, then the subscriber application has two choices:

- Having content *pushed* to him based on an event, such as whenever content is updated
- Having content *pushed* to him based on a content update time interval, in which content is periodically updated, for example every hour based on how the scheduler is set

1.5.5 ICE Request Runtime Flow

The ICE request runtime flow is shown in Figure 1–5 and described in more detail in the steps that follow.

Figure 1–5 ICE Request Runtime Flow



1. A subscriber application issues an ICE request (for example, get catalog, subscribe, get content). The appropriate delivery operation module (for example, HTTP, SMTP) listens for the request and forwards it to the request manager.
2. The request manager receives the request and verifies the subscriber application credentials with the subscriber manager (not shown in figure). All subscriber application profiles, including preferred communication channels and contact information, are managed by the subscriber manager. Once credentials are verified, the request is dispatched to the subscription manager, affiliates manager, and message manager.

3. Message manager, subscription manager, and affiliates manager operations include the following:

   a. The message manager parses the ICE request payload, converting the request to an internal format used by the subscription manager and the affiliates manager.

   b. Upon receiving a subscribe request, the subscription manager works in combination with the affiliates manager to approve the request. A negotiation process defined by ICE standards might be started if there is any disagreement with the service terms in that subscriber application's request, such as delivery update frequencies, and so forth. The subscription manager generates the subscription once both the subscriber application and Syndication Server (the syndicator) reach a mutual agreement. Also, at this time a unique subscription ID is assigned for future reference. After establishing the subscription, the subscription manager also stores the subscription ID persistently in the Syndication Server repository.

      If the subscription contains any time-based *push* delivery rules, an instance of the scheduler is instantiated based on the time interval value. Upon establishing the subscription, an instance of the scheduler is created to carry out the time-based *push* delivery. During this instantiation phase, the time interval and runtime parameters (target notification URL and the associated subscription ID) are set. The container notifies the scheduler instance periodically based on the preset interval.

      Upon a get-content request, the content offer, delivery policies, and business terms for the subscription are retrieved by the subscription manager. The content offer details are used by the affiliates manager to find the appropriate content provider adaptor to transform the subscriber application's request into a format suitable for the specific content provider.

   c. Based on the type of request (for example, get catalog, subscribe, or get content) that is made by the subscriber application, the affiliates manager invokes the specified content provider adaptor.

4. The transformed request is sent to the content provider, and the corresponding response is transformed to a well-defined XML schema format. The response is then packaged as an ICE payload and delivered to the subscriber application for reuse.

2

Configuration

To manage Oracle9*i*AS Syndication Server, use the Syndication Server Manager, a component of Oracle Enterprise Manager. This chapter describes how to configure Syndication Server.

2.1 Configuring Syndication Server

This section describes how to configure Syndication Server.

Configuring Syndication Server

The following instructions configure Syndication Server:

1. Configure the Syndication Server global properties by launching the Oracle Enterprise Manager (OEM) Syndication Server Manager and select the Global Properties option. Modify each of the global properties to meet the your own requirements. See Section 4.1.6 for more information on managing Syndication Server global properties.
2. Check to see that the following preregistered sample content providers `cpa001` and `cpa002` are registered.

   Start OEM Syndication Server Manager and select the Content Providers option. See Section 4.1.3 for more information on managing content providers.

   CPA001: Sample Database Content Provider (eShopAudio)

   CPA002: Sample Filesystem Content Provider (File)
3. Execute the following shell script for each content provider that you want to configure. In this case, you want to configure content providers: `cpa001` and `cpa002`.

   <$IAS_HOME>`/syndication/bin/cpaconfig.sh`

Assume that the directory exists under the *<$IAS_HOME>* directory and that the current Syndication Server directory is *<$IAS_HOME>*/`syndication`.

3

Syndication Server Development

This chapter describes the development flow for setting up a content syndication system following the installation, deployment, and configuration of Oracle9*i*AS Syndication Server. The tasks include:

- Creating Additional Required Packages (see Section 3.1)
- Registering subscriber applications (see Section 3.2.1)
- Registering content providers (see Section 3.2.2)

3.1 Creating Additional Required Packages

This section describes two additional types of required packages that you must create manually and place as a zip file in either a local file directory or in a remote file location that is accessible using an HTTP URL.

- Content provider description package

  Before you can register a content provider, you must manually create this package.

- Subscriber application description package

  Before you can register a subscriber application, you must manually create this subscriber application description package.

Section 3.1.1 and Section 3.1.2 describe each of these packages in greater detail.

3.1.1 Content Provider Description Package

This section describes the content provider description package that is required in order to register a content provider. The content description package contains the name, brief description of the content provider, contact e-mail address, telephone

number, postal address, and the location of the business agreement document. This package must be manually created. Instructions for creating these packages are described in this section. The Syndication Server Manager for Oracle Enterprise Manager will support automatically creating this package in a future release.

An example content provider description package (`eShop_CP.zip`) is supplied with Syndication Server. The contents of the manifest file `/oracle/syndication/server/cp_eshop.xml` points to the following `cp_eshop.xml` file, whose content is shown in the following example:

```
<?xml version="1.0"?>
<CONTENT_PROVIDER xmlns="http://www.oracle.com/ds/2000/SERVICE_DESCRIPTOR/CONTENT_
PROVIDER">
   <NAME>Sample Affiliate</NAME>
   <DESCRIPTION>Sample Affiliate</DESCRIPTION>
   <EMAIL>cpadmin@company.com</EMAIL>
   <PHONE>(123)456-7890</PHONE>
   <ADDRESS>one oracle drive</ADDRESS>
   <BUSINESS_TERM>/oracle/syndication/server/bizTerm.pdf</BUSINESS_TERM>
</CONTENT_PROVIDER>
```

For each content provider that you want to register as a content provider, you must create a content provider package zip file, exactly as shown in this sample content provider package zip file.

3.1.2 Subscriber Application Description Package

This section describes the subscriber application description package that is required before you can register a subscriber application. The subscriber application description package contains the name, brief description of the subscriber application, contact e-mail address, telephone number, postal address, and the location of the default URL for the subscriber application. This package must be manually created. The Syndication Server Manager for Oracle Enterprise Manager will support automatically creating this package in a future release.

An example subscriber application description package (`foo_sub.zip`) is supplied with Syndication Server. The contents of the manifest file `/oracle/syndication/server/sub_foo.xml` points to the following `sub_foo.xml` file, whose content is shown in the following example:

```
<?xml version="1.0"?>
<SUBSCRIBER xmlns="http://www.oracle.com/ds/2000/SERVICE_DESCRIPTOR/SUBSCRIBER" >
   <NAME>Sample Subscriber</NAME>
   <DESCRIPTION>Sample Subscriber</DESCRIPTION>
   <EMAIL>chan@us.oracle.com</EMAIL>
```

```
   <PHONE>(123)456-7890</PHONE>
   <ADDRESS>one oracle drive</ADDRESS>
   <DEFAULT_URL>http://localhost/syndclient/OSC</DEFAULT_URL>
</SUBSCRIBER>
```

For each subscriber application that you want to register as a subscriber, you must create a subscriber application description package zip file, exactly as shown in this sample subscriber application description package zip file.

3.2 Registering Subscriber Applications and Content Providers

After creating the content provider description package described in Section 3.1.1 and the subscriber application description package described in Section 3.1.2, you can begin to register and manage subscriber applications and content providers.

3.2.1 Registering and Managing Subscriber Applications

See Section 4.1.1 for information on how to register and manage subscriber applications using the Manage Subscribers management task that is part of the Syndication Server Manager for OEM administration tool.

3.2.2 Registering and Managing Content Providers

See Section 4.1.3 for information on how to register and manage content providers using the Manage Content Providers management task that is part of the Syndication Server Manager for OEM administration tool.

4

Syndication Server Administration

This chapter provides a brief overview of the Syndication Server Manager for Oracle Enterprise Manager (OEM) used for Syndication Server administration tasks.

4.1 Syndication Server Manager for Oracle Enterprise Manager

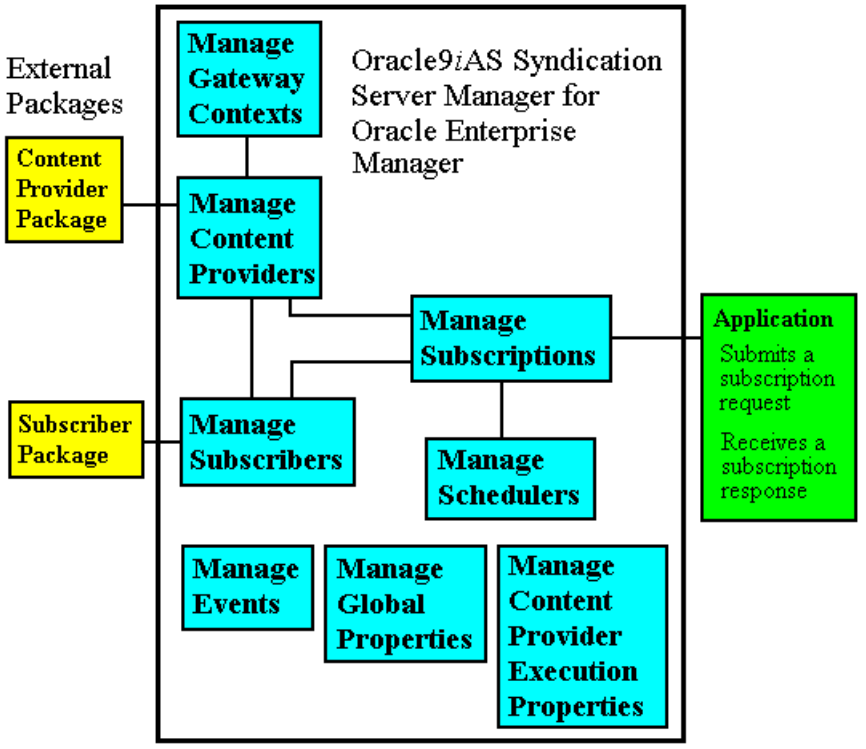
Syndication Server Manager, a Web-based administration tool available through Oracle Enterprise Manager lets you administer Oracle9*i*AS Syndication Server. Syndication Server administration tasks are organized into the following management tasks:

- Managing Subscribers (applications)
- Managing Subscriptions
- Managing Content Providers
- Managing Schedulers
- Managing Gateway Contexts
- Managing Global Properties
- Managing Events
- Managing Content Provider Execution Properties

Figure 4–1 provides an overview of the Syndication Server management tasks that can be performed, and the relationship among some of these tasks. In addition, the relationship of some tasks performed at runtime, such as creating subscriptions and schedulers, are shown to provide a better understanding of how Syndication Server works, along with some of the required packages needed to register content providers and subscriber applications.

Each of these tasks is described in more detail in Section 4.1.1 through Section 4.1.8.

Figure 4–1 Overview of Oracle9iAS Syndication Server Manager for OEM



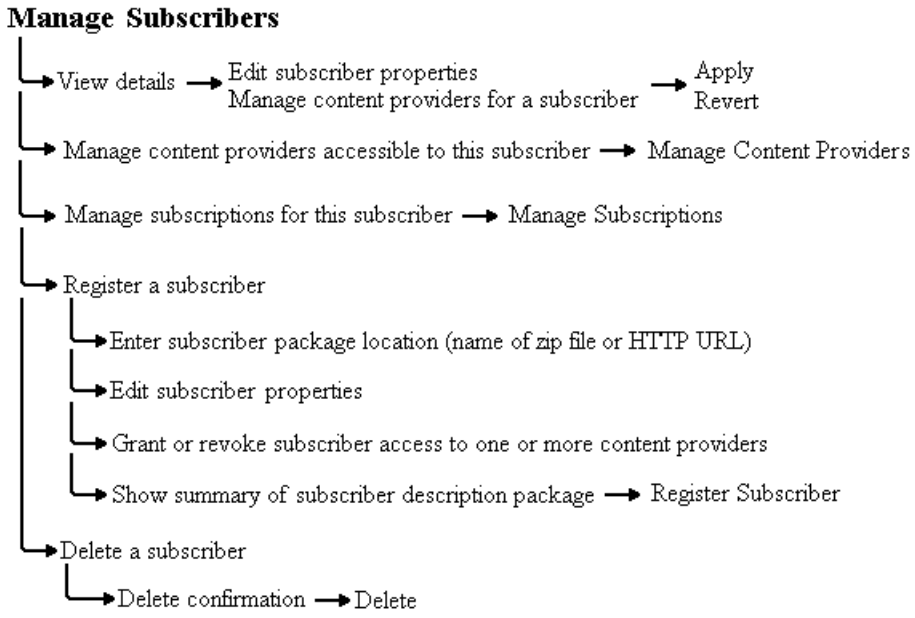
4.1.1 Managing Subscribers (Applications)

From the Subscribers window, you can:

- Edit subscriber properties
- Manage content providers
- Manage subscriptions
- Register a subscriber
- Delete a subscriber

Figure 4–2 shows a flow diagram of the Manage Subscriber tasks that you can perform from the Subscribers window.

Figure 4–2 Flow Diagram of the Manage Subscriber Tasks



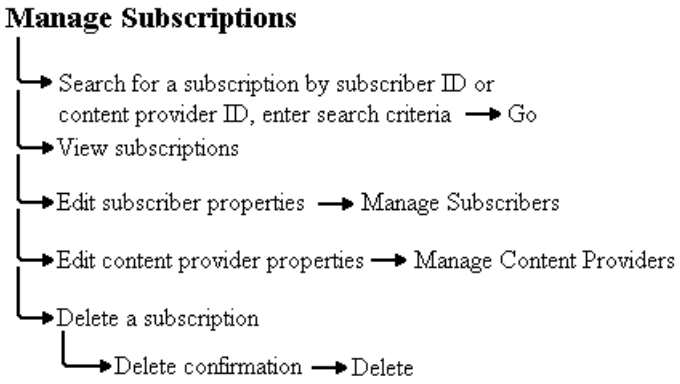
4.1.2 Managing Subscriptions

From the Subscriptions window, you can:

- Search for subscriptions by subscriber ID or content provider ID and enter the search criteria
- View subscriptions
- Edit subscriber properties
- Edit content provider properties for that subscription
- Delete a subscription for a selected subscriber

Figure 4–3 shows a flow diagram of the Manage Subscription tasks that you can perform from the Subscriptions window.

Figure 4–3 Flow Diagram of the Manage Subscription Tasks



4.1.3 Managing Content Providers

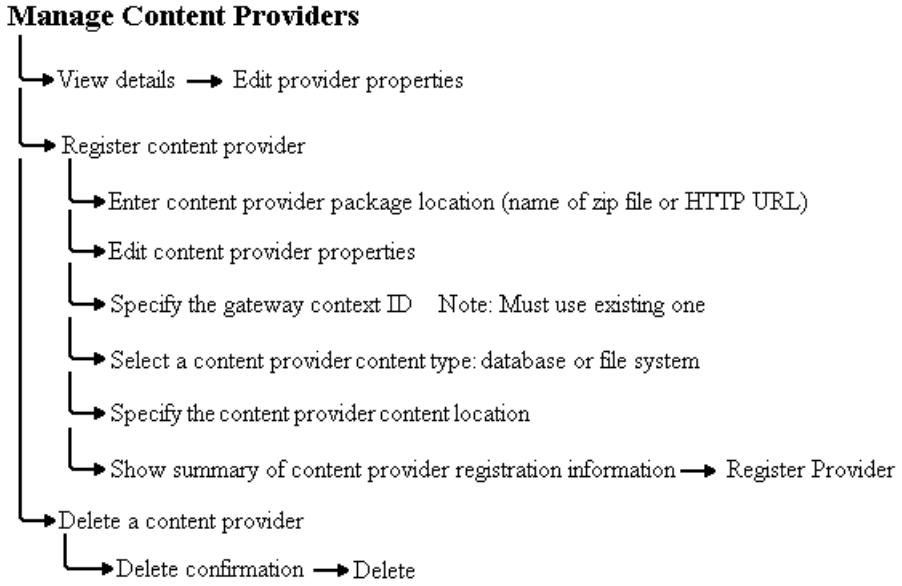
From the Content Providers window, you can:

- Edit the following content provider properties:
  - Content provider name
  - Content provider description
  - Content provider postal address
  - Content provider e-mail address
  - Content provider telephone number
- Register each content provider

  You must register each content provider using the Register Content Provider wizard available on the Content Providers window.
- Delete registered content providers

Figure 4–4 shows a flow diagram of the Manage Content Provider tasks that you can perform from the Content Providers window.

Figure 4–4 Flow Diagram of the Manage Content Provider Tasks



4.1.4 Managing Schedulers

After Syndication Server approves a subscription, it creates a scheduler (a DBMS_JOB instance residing in an Oracle database) for each delivery rule that has the delivery mode set to *push* and sends it to the scheduler.

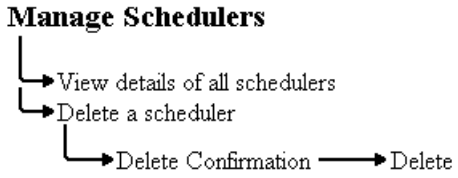
Information for each scheduler includes:

- Scheduler ID -- The ID of the scheduler.
- Subscription ID -- The ID of the subscription.
- Content Provider ID -- The Universal Unique Identifier (UUID) or ID assigned to the content provider at registration time.
- Notification URL -- The Syndication Server URL used to listen for the scheduler.
- Subscriber PUSH URL -- The URL of the subscriber to which content is *pushed.*
- Next Awake Date -- The next date and time when the scheduler will be triggered.

- Status (broken) -- The status (No or Yes) indicating whether or not the scheduler is broken. Yes means the scheduler is broken and could not schedule and execute the task for which it was created.
- Creation Date -- The date when the scheduler was created.
- Interval (sec) -- The time interval in seconds between scheduled deliveries of content.

Figure 4–5 shows a flow diagram of the Manage Scheduler tasks that you can perform from the Schedulers window.

Figure 4–5 Flow Diagram of the Manage Scheduler Tasks



4.1.5 Managing Gateway Contexts

A gateway context represents a content provider's connection with all its properties to a content provider instance. Each content provider registers all of its content services with one content provider instance, which is defined by its gateway context. For better performance and load balancing, a content provider should register its content services with different content provider instances, that is, it should consider using different Gateway Context IDs.

Gateway Context Properties include the following information:

- Gateway Context ID
- Service User Name
- Service User Password
- Service Driver -- for example, the Direct Driver -- oracle.ds.driver.DSDirectDriver.
- Connection URL

  Enter a connection URL using the following syntax:

```
jdbc:oracle:oci8:@<tns-name-entry-to-where-content
provider-instance-is-installed>
```

or

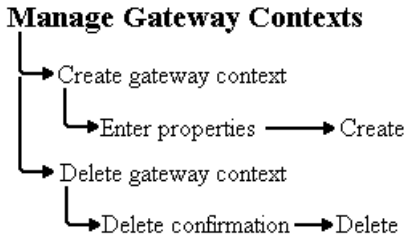
```
jdbc:oracle:thin@<your-host-name>:<your-port-number>:<tns-name-entry-to-wher
e-content-provider-instance-is-installed>
```

The connection URL represents your database transport network service name (tnsname) or service identifier for using either the oci8 driver or the thin driver. This connection URL connects you to the instance to where the content provider registers all of its content services.

- Connection Pool Size -- The default is 5.

Figure 4–6 shows a flow diagram of the Manage Gateway Contexts tasks that you can perform from the Gateway Contexts window.

Figure 4–6 Flow Diagram of the Manage Gateway Contexts Tasks



4.1.6 Managing Global Properties

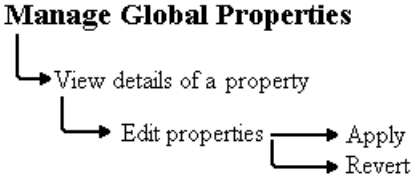
The Syndication Server global properties that you can manage include:

- HTTP_CERT_PATH -- The directory containing the Oracle Wallet certificates for handling secure transactions.
- PUSH_HANDLER_URL -- The Web-based handler or HTTP URL that accepts notifications from schedulers for *push* content delivery.
- OSS_EMAIL_SERVER -- The e-mail server URL configured to send the assigned Universal Unique ID (UUID) to the registrant.

- OSS_LOGGING_LEVEL -- The level of detail to capture in logging events. Levels are 0: Terse, 1: Verbose, and 2: Trace.
- proxySet -- The flag that indicates whether the proxy server is enabled or disabled. Values are true or false.
- OSS_LOGGING_ENABLE -- The switch that indicates whether system logging is enabled or disabled. Values are true or false.
- OSS_ADMIN_EMAIL -- The e-mail URL for the sender field in the notification e-mail.
- OSS_EVENTQUEUE_HOSTURL -- The Oracle database URL to where the event message queue resides.
- OSS_PUSH_GWCTXTX_ID -- The gateway context ID used for executing *push* content delivery within Syndication Server.
- proxyPort -- The proxy server port. The default is 80.
- proxyServer -- The proxy server URL based on your own configuration.

Figure 4–7 shows a flow diagram of the Manage Global Properties tasks that you can perform from the Global Properties window.

Figure 4–7 Flow Diagram of the Manage Global Properties Tasks



4.1.7 Managing Events

Event information includes:

- Timestamp -- The timestamp of the event.
- Type of event -- Currently only one type of event occurs: OSS_EV_LOGGING.
- Consumer ID -- The Universal Unique ID (UUID) of the consumer.
- Consumer description -- A description of the consumer operation.

- Service ID -- The name of the Syndication Server service, if applicable.
- Message type -- The type of message logged, such as CATALOG, REQUEST, PULL, ONSUBSCRIBE, and CANCEL.
- Status -- The status of the event, SUC means it was successful or FAIL means it failed.

Events are logged at three levels depending on first, whether the OSS_LOGGING_ENABLED global property is set to true or enabled, and second, the level of logging specified for the OSS_LOGGING_LEVEL global property, which is:

- 0 =TERSE

  This is the minimal level of logging Syndication Server events and returns the status of the event as SUC (successful) or FAIL (failed).

- 1 = VERBOSE

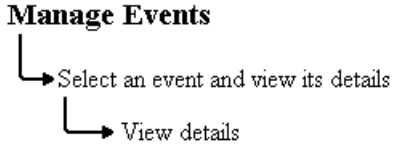
  This logs major steps of Syndication Server events, including the event status as successful or failed, and any warning messages that were generated.

- 2 = TRACE

  This is the most detailed level of logging of Syndication Server events and returns the major steps of Syndication Server events, including the event status as successful or failed, any warning messages that were generated, and detailed trace information to help debug problems.

Figure 4–8 shows a flow diagram of the Manage Events tasks that you can perform from the Events window.

Figure 4–8 Flow Diagram of the Manage Events Tasks



Before you start the event listener daemon to begin listening for Syndication Server events, you must enable system logging from Manage Global Properties so these events can be written to the log file. Set OSS_LOGGING_ENABLE to true to enable

system logging, then start the event listener daemon. To start the event listener daemon, run the following command:

```
In UNIX:
$ORACLE_HOME/syndication/bin/ossmon -u dsgateway -e start
In Windows:
$ORACLE_HOME\syndication\bin\ossmon -u dsgateway -e start
```

The event listener daemon displays the following information as it starts up:

```
Oracle Syndication Server Monitor Starting up...
OSS_EV_LOGGING: Init of Oracle AQ...
OSS_EV_LOGGING: Init of Oracle AQ... Done.
OSS_EV_LOGGING: Monitor added.
OSSEvM.init: All Monitors initialized
Inside run() WHILE-Loop
Oracle Syndication Server Starting up... Done.
Right before Dequeue...
```

4.1.8 Managing Content Provider Execution Properties

You can manage the following content provider execution properties:

- Internet access from within a firewall
- Content provider response caching
- Debug logging level

Internet Access

If you are accessing the Internet from within a firewall, select Access the Internet using a proxy server, then enter the address of your proxy server and its port number.

Content Provider Response Caching

Caching is enabled or disabled for all packages at the content provider execution level. Note that this setting takes precedence over any response caching parameters set in the package.

To cache all responses in the Oracle database at the execution level, select Enable caching for all content provider responses.

Caching is done by storing content provider responses indexed by their corresponding content provider requests. When performing cache lookup

operations, a full XML comparison is made between the incoming request and the requests used to index the responses. During this comparison, white space differences can be ignored or noted. Choosing to ignore white spaces implies a looser match and therefore a more frequent use of the cache, thereby increasing performance from the application's perspective.

To ignore white space in service requests or more frequent use of the cache, select Ignore white spaces in content provider requests for cache lookup operations.

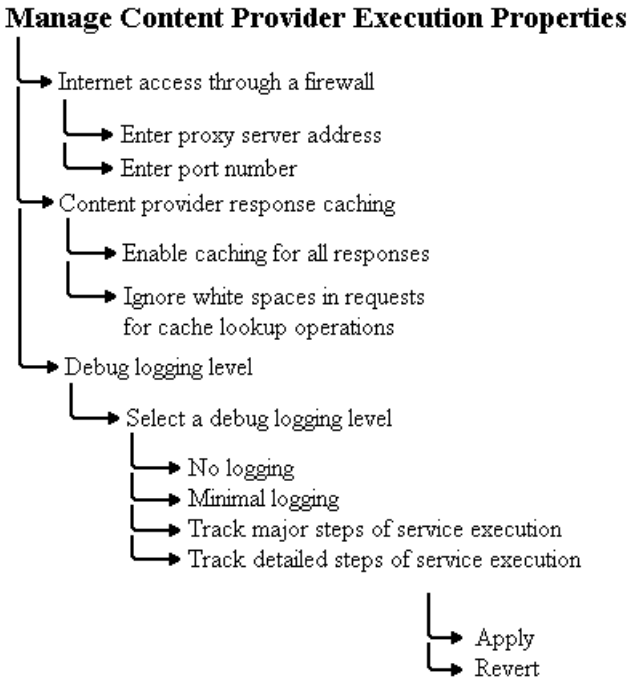
Debug Logging Level

To select the level of debug logging you want written to the Web server log file, go to the pull-down menu and make a selection. The options are:

- No logging
- Minimal logging -- returns the status of service execution as successful or failed
- Track major steps of service execution -- returns the status of service execution as successful or failed, the major steps of service execution, and any warning messages that were generated
- Track detailed steps of service execution -- returns the status of service execution as successful or failed, the major steps of service execution, any warning messages that were generated, and detailed trace information to help debug problems

Figure 4–9 shows a flow diagram of the Manage Content Provider Execution Properties tasks that you can perform from the Content Provider Execution Properties window.

Figure 4–9 Flow Diagram of the Manage Content Provider Execution Properties Tasks



A

Subscriber Application Development Kit

For a subscriber application to interact with Oracle9*i*AS Syndication Server, it must be ICE-compliant. If your subscriber application is a non ICE-compliant application, you must use the Oracle-supplied subscriber application development kit that has an API client library that will make your application ICE-compliant.

Section A.1 describes ICE-compliant applications and Section A.2 describes how to make your non ICE-compliant applications, ICE-compliant.

A.1 ICE-Compliant Applications

If your subscriber application is already ICE-compliant, then it can interact with Oracle9*i*AS Syndication Server with no modifications.

A.2 Non ICE-Compliant Applications

The Subscriber Application Development Kit (SADK) has a client library to assist subscriber application developers in building their own ICE-compliant applications.

The client library is written in Java and contains the following packages:

- Package - oracle.syndication provides the interface for the constants of ICE 1.1 status code. For example, constant ICE200 stands for Ok. Each ICE response payload must contain a status code section and the constant for the status code should be from the ICE status code.
- Package - oracle.syndication.ice.payload abstracts all XML formats defined by ICE 1.1 and present them as Java objects. Each Java class stands for one XML element defined in ICE 1.1. For example, Ice_sender.java corresponds to the ice-sender element in a regular ICE payload in XML
- Package - oracle.syndication.server.message.ice has all the implementations for interfaces defined inside the package oracle.syndication.server.message. ICE-

MessageManager is responsible for marshalling / unmarshalling messages in both directions in ICE 1.1 format. For instance, ICEMessage can construct a Java object of Ice_payload out of the incoming server response message.

- Package - oracle.syndication.server.transport.http supports the communication from the subscriber application to Oracle9*i*AS Syndication Server over the HTTP layer. The interface of OSSTransProtocolManager is defined inside the package of oracle.syndication.server.transport.
- Package - oracle.syndication.util.iso8601 provides the conversion for the date and time between an ISO8601-compliant string and a Java object class.
- Package - oracle.syndication.client.local defines and implements all required functions for a subscriber application. It also provides a sample implementation for the ICEClient and the ICEClientOfferManager.

Oracle Corporation recommends that the subscriber application implement its own offer manager from the interface ICEClientOfferManager in its preferred way. Offers and subscribers for the subscriber application should be kept persistently.

The ICE client library is available after installing Oracle9*i*AS Syndication Server. This ICE client library allows you to communicate with any ICE 1.1-compliant syndication server, and in particular you can use this client to communicate with Oracle9*i*AS Syndication Server.

The client library can be found under the `demo/client/lib/` directory for UNIX or the `demo\client\lib\` directory for Windows as the file `syndclient.jar`. The API documentation describes how to use this client library and can be found on the Oracle9*i*AS Documentation Library CD-ROM as API Reference (Javadoc) under Oracle9*i*AS Syndication Server, which is located under the Portals tab.

In addition, included with this release of Syndication Server is a sample client code build on top of the client library that builds a servlet that exposes a Web interface for the client. The `Makefile` for this servlet can be found in the `demo/client/src` directory on UNIX or `demo\client\src` directory on Windows.

B

Frequently Asked Questions

A text file containing a list of frequently asked questions is available online after installing Oracle9*i*AS Syndication Server.

This text file can be found at:

On UNIX systems:

$ORACLE_HOME/syndication/doc/FAQ.txt

On Windows systems:

$ORACLE_HOME\syndication\doc\FAQ.txt

B.1 Managing Job Queues

Refer to the Managing Job Queues chapter in *Oracle9i Database Administrator's Guide* to make sure that you have enabled processes for executing jobs. The two issues to be concerned about are:

- The value for the dynamic initialization parameter, JOB_QUEUE_PROCESSES. For example,

  ```
  sqlplus "sys/change_on_install as sysdba"
  ALTER SYSTEM SET JOB_QUEUE_PROCESSES=20
  ```

- Disabling a restricted session in an Oracle9*i* instance using the initialization parameter, DISABLE RESTRICTED SESSION. For example,

  ```
  sqlplus "sys/change_on_install as sysdba"
  ALTER SYSTEM DISABLE RESTRICTED SESSION
  ```

Because both of these initialization parameters are dynamic, any changes you make will take effect immediately without having to stop and restart the database instance.

C

Web Sites

The following is a list of Web sites that you may find useful during the use or development of Syndication Server services.

- W3C Extensible Markup Language (XML) 1.0 (Second Edition) Specification

  `http://www.w3.org/TR/2000/WD-xml-2e-20000814`

- W3C Extensible Stylesheet Language (XSL) Specifications

  `http://www.w3.org/TR/xsl/`

- W3C XSL Transformations (XSLT) Specifications

  `http://www.w3.org/TR/1999/PR-xslt-19991008`

- W3C XML Schema Specifications Part 0: Primer

  `http://www.w3.org/TR/xmlschema-0/`

- W3C XML Schema Specifications Part 1: Structures

  `http://www.w3.org/TR/xmlschema-1/`

- W3C XML Schema Specifications Part 2: Datatypes

  `http://www.w3.org/TR/xmlschema-2/`

- W3C Namespaces in XML

  `http://www.w3.org/TR/1999/REC-xml-names-19990114/`

- W3C Extensible HyperText Markup Language (XHTML) Specifications

  `http://www.w3.org/TR/1999/xhtml1-19990505/lastCallDiff`

- W3C HTTP 1.0 Specification

  `http://www.w3c.org/Protocols`

- Information Content and Exchange (ICE) Protocol Specifications

  `http://www.icestandard.org/`

- Information Content and Exchange (ICE) Protocol V1.1 Specification

  `http://www.icestandard.org/spec/SPEC-ICE1.01-20000511.html`

D

Syndication Server Error Messages

The following sections describe the error messages of Oracle9*i*AS Syndication Server.

D.1 Syndication Server Errors

The following messages are associated with Syndication Server.

D.1.1 2*nn*: Success Status Codes

SYND-200, OK

Cause: The operation completed successfully.

Action: No action need be taken.

SYND-201, Confirmed

Cause: The operation is confirmed.

Action: No action need be taken.

SYND-202, Package sequence state already current

Cause: A subscriber requested a package update but the subscriber is already in the current package sequence state, therefore there are no updates at the moment.

Action: No action need be taken.

D.1.2 3*nn*: Payload Level Status Codes

SYND-300, Generic catastrophic payload error

Cause: A generic status code indicating an inability to comprehend the received payload.

Action: Check the request payload and resend it; if the problem persists, have your Syndication Server administrator check the status of the requester's account.

SYND-301, Payload incomplete or cannot parse

Cause: The payload sent is severely garbled and cannot be parsed possibly because it is not an XML payload. Expecting an XML payload and received a payload that was not XML.

Action: Resend the payload and be sure it is an XML payload.

SYND-302, Payload not well formed XML

Cause: The payload sent is recognizable as XML, but is not well formed per the definition of XML.

Action: Ask your Syndication Server administrator to fix the corresponding syndication operation module.

SYND-303, Payload validation failure

Cause: The payload failed validation according to the DTD.

Action: Check the request payload. It must conform to the specified request DTD provided by Syndication Server.

SYND-304, Temporary responder problem

Cause: The responder is either too busy, an update is in progress, and so forth. Eventually an identical retry request might succeed.

Action: Check the requester's local message system for any notification of system downtime, or try again later.

SYND-320, Incompatible version

Cause: The communication protocol used in the request is not supported.

Action: If you use the Syndication Server client library, check the list of protocol versions it supports and then contact Oracle Customer Support Services. If you do not use the Syndication Server client library, contact your local system administrator for help.

SYND-331, Failure fetching external data

Cause: The receiver could not follow an external reference (URL) given to it by the sender as an external entity reference. Syndication Server must not reply with this code.

Action: Double check the content offer reference link and all of its related access policies, such as certificates for an HTTP or HTTPS approach.

SYND-390, Payload temporary redirect

Cause: The payload is being redirected temporarily using the supplied new transport communication endpoint. In ICE / HTTP, this means a new URL is supplied.

Action: Contact Oracle Customer Support Services.

SYND-391, Payload permanent redirect

Cause: The payload is being redirected permanently using the supplied new transport endpoint. In ICE / HTTP, this means a new URL is supplied.

Action: Contact Oracle Customer Support Services.

D.1.3 4*nn*: Request Level Status Codes

SYND-400, Generic request error

Cause: A generic status code indicating an inability to comprehend the request.

Action: Check the request payload for errors and resend it.

SYND-401, Incomplete or cannot parse

Cause: The request sent is severely garbled and cannot be parsed.

Action: Check the request payload for errors and also make sure you close the communication channel only after the request is completely sent out.

SYND-402, Not well formed XML

Cause: The request sent is unrecognizable XML, but is not well formed per the definition of XML.

Action: Check the request payload for errors.

SYND-403, Validation failure

Cause: The request failed validation according to the DTD.

Action: Make sure the request payload conforms to the request DTD provided by Syndication Server.

SYND-405, Unrecognized sender

Cause: The sender does not have permission to access the current server.

Action: Contact your Syndication Server administrator to verify sender information.

SYND-406, Unrecognized subscription

Cause: Either the subscription ID is not correct, or the subscription is not in a consistent state.

Action: Contact your Syndication Server administrator to verify sender information.

SYND-407, Unrecognized operation

Cause: The request type is not supported at the server side.

Action: Make sure the operation you want to perform is supported for Syndication Server.

SYND-408 Unrecognized operation arguments

Cause: The request argument is not supported at the server side.

Action: Check the request payload against the request DTD provided by Syndication Server. If the error persists, contact Oracle Customer Support Services.

SYND-409, Not available under this subscription

Cause: The requester has referred to something not covered by the subscription referenced in the request.

Action: Resubscribe with additional conditions.

SYND-410, Not found

Cause: A generic error for being unable to find something.

Action: Check the error message for the detail description.

SYND-411, Unrecognized package sequence state

Cause: The package sequence identifier supplied by the sender is not understood by the receiver.

Action: Check the payload for errors. If the error persists, contact Oracle Customer Support Services.

SYND-412, Unauthorized

Cause: The subscriber does not have permission to access Syndication Server or a certain set of content providers.

Action: Contact your Syndication Server administrator.

SYND-413, Forbidden

Cause: Access violation according to the subscriber's profile.

Action: Contact your Syndication Server administrator.

SYND-414, Business term violation

Cause: The current operation or the content of the current request does not comply with the business terms between the subscriber and Syndication Server.

Action: Review the business terms for the subscriber account and adjust the request.

SYND-420, Constraint failure

Cause: A violation or inconsistency for the subscriber's request for content occurred from Syndication Server according to constraints specified in the subscriptions.

Action: Review the constraint for the subscription and adjust the subscriber's request payload.

SYND-422, Schedule violation, try again later

Cause: The request was made at an incorrect time.

Action: Issue the request for a package update within the agreed-upon timing window.

SYND-430, Not confirmed

Cause: A generic error indicating the operation is not confirmed.

Action: Check the subscriber's local system setting to send confirmation. If the error persists, contact Oracle Customer Support Services.

SYND-431, Error fetching external data

Cause: The receiver could not follow an external reference (URL) given to it by the sender.

Action: Check the subscriber's local system settings to enable such action.

SYND-440, Sorry

Cause: As part of the negotiation process, this response indicates the proposal is unacceptable for some reason.

Action: Review the subscription request and make necessary modifications; then resend the request.

SYND-441, Counter proposal

Cause: As part of the negotiation process, this response indicates an updated offer or a counter proposal.

Action: Review the counter offer subscription request and make necessary modifications, then send the response.

SYND-442, Renegotiation in progress

Cause: As part of the renegotiation process, this response indicates that renegotiation is in progress.

Action: Proceed to the renegotiation process.

SYND-443, Offer acknowledged but deferred

Cause: Used during parameter negotiation. The receiver of an offer cannot evaluate the offer without human intervention. The sender may try again later with the offered parameters and the same subscription ID. The receiver is expected to respond with a new offer (reflecting the result of the intervention) at some later time.

Action: After waiting for a reasonable period of time, if there is no response, contact your Syndication Server administrator to see if there was a network problem or system problem, which might explain why no response was received. If there were no obvious network or system problems, then the receiver may either contact the sender directly and come to some agreement, at which time the sender will send out a new offer, or the sender may resend the previous offer at a later time.

SYND-450, Range type invalid

Cause: Used during parameter negotiation. A protocol-specific RANGE of type numerical, lexical, time, or enumeration, has been incorrectly specified.

Action: Review the corresponding sections inside the request payload and make corrections.

SYND-451, Span selection out of range

Cause: Used during parameter negotiation. The value selected in a span is less than the minimum or greater than the maximum.

Action: Review the request payload and make the correction.

SYND-452, Selection error

Cause: Used during parameter negotiation. The number of items selected in an enumeration does not match the `select` attribute. For example, the `select` attribute may require one or more items to be selected but the number selected in the response was zero.

Action: Review the request payload and make the correction.

SYND-453, Enumeration selection empty or invalid

Cause: Used during parameter negotiation. The number of selectable `ice-enum-item` elements is not compatible with the `select` attribute. For example, the `select` attribute might indicate that at least one `ice-enum-item` must be selected, but the actual number of items available is less than the number of `ice-enum-item` elements in the enumeration.

Action: Review the request payload and make the correction.

SYND-454, Span invalid

Cause: Used during parameter negotiation. A span has values that violate its structure. For example, a span point must not have a value less than the span minimum nor larger than the span maximum.

Action: Review the request payload and make the correction.

SYND-456, Time formats incompatible for comparison

Cause: Used during parameter negotiation. Two time values to be compared during negotiation are in incompatible formats. For example, a duration is being compared with an ICE date or time.

Action: Review the request payload and make the correction.

D.1.4 5*nn*: Implementation Errors and Operational Failures

SYND-500, Generic internal responder error

Cause: This is a catch-all response for general problems.

Action: Contact your Syndication Server administrator.

SYND-501, Temporary responder problem

Cause: The responder is too busy, an update is in progress, and so forth.

Action: Eventually, an identical retry request might succeed.

SYND-503, Not implemented

Cause: The service does not implement the requested operation.

Action: Make sure the operation you want to perform is supported by Syndication Server.

D.1.5 6*nn*: Pending State

SYND-601, Unsolicited message must be processing now

Cause: The syndicator has unsolicited messages to send to the subscriber, and the subscriber has not yet requested them.

Action: Issue the request for pending packages to Syndication Server.

SYND-602, Excessive confirmations outstanding

Cause: The syndicator had requested confirmation of package delivery, and now refuses to perform any additional operations until the subscriber supplies the confirmations (positive or negative).

Action: Issue the confirmation for the requested package delivery.

SYND-603, No more confirmations to send

Cause: Syndication Server requests a package delivery confirmation that the subscriber believes has already been sent out.

Action: No action is needed.

SYND-604, No more unsolicited messages

Cause: The subscriber sent an ice-unsolicited-now message but the syndicator has no unsolicited messages to send.

Action: Check the subscriber's local system; if the inconsistencies still persist, contact Oracle Customer Support Services.

D.2 ICE Errors

The following messages are associated with the Syndication Server implementation of the ICE Version 1.1 specification.

D.2.1 7*nn*: Local Use Codes - Reserved for Use by the Local ICE Implementation

SYND-701, Failed to process subscription

Cause: This error happens while Syndication Server is processing the subscription request.

Action: Review the request payload and correct the errors.

SYND-702, Subscription already exists

Cause: The identical subscription to the current one already exists.

Action: Abandon the current request or regenerate the subscription request with different content, such as the ID.

SYND-703, Content provider already exists

Cause: The same content provider is already registered with Syndication Server. This is a Syndication Server administrator error message.

Action: Abandon the current transaction.

SYND-704, Content provider not found

Cause: Invalid content provider ID or the content provider has been removed. This is a Syndication Server administrator error message.

Action: Contact your Syndication Server administrator.

SYND-705, Content provider adaptor not found

Cause: Syndication Server could not locate the source to instantiate the adaptor.

Action: Contact your Syndication Server administrator to determine why there is an inconsistent state for the content provider.

SYND-706, Failed to load the Content Provider Adaptor class

Cause: The adaptor cannot be instantiated by Syndication Server.

Action: The content provider adaptor is corrupted. Contact your Syndication Server administrator.

SYND-707, I/O exception when trying to send a request

Cause: The transport manager failed to send out a payload due to protocol exceptions or because Syndication Server encountered an internal error. There may be a problem with your client library.

Action: Contact Oracle Customer Support Services.

SYND-708, HTTP response error

Cause: When using HTTP/S as the concrete network protocol, it indicated that some network errors happened. There may be a problem with your client library.

Action: Check and make sure the destination URL is correct; contact Oracle Customer Support Services.

SYND-709, Invalid content type for the response

Cause: When using HTTP/S as the concrete network protocol, the content type is not defined.

Action: Contact Oracle Customer Support Services.

SYND-710, Failed to spawn a scheduler

Cause: During the subscription process, if the subscriber asks for auto push delivery and Syndication Server cannot start up a scheduler to handle it, an exception will be returned.

Action: Contact your Syndication Server administrator.

SYND-711, Subscriber already exists

Cause: The subscriber account has already been created on Syndication Server. This is an administrator error message.

Action: Abandon the current registration process. Use the existing subscriber.

SYND-712, Subscriber not found

Cause: Cannot locate the subscriber by the given ID.

Action: Provide a valid subscriber ID.

SYND-713, There is no content provider associated with the current subscriber

Cause: The current subscriber has not been granted permission to access any content provider.

Action: Contact your Syndication Server administrator.

SYND-714, Subscription already expired

Cause: You tried to access content from an expired subscription.

Action: Resubscribe to the content offer.

SYND-715, Scheduler not found

Cause: Syndication Server internal error. The given scheduler has already been removed.

Action: Validate the scheduler ID by using the OSSAdmin utility.

SYND-716, The specified content provider is not accessible to the current subscriber

Cause: Permission is not granted to the subscriber to access the specified content provider.

Action: Verify the content provider ID is correct; if the problem persists, contact your Syndication Server administrator.

SYND-720, Event message exception: Invalid event message format; cannot be parsed

Cause: The event message format is not valid and cannot be parsed.

Action: Contact your Syndication Server administrator.

SYND-721, Event message exception: Failed to initialize the AQ/JMS topic

Cause: The AQ/JMS topic could not be initialized.

Action: Contact your Syndication Server administrator.

SYND-722, Event message exception: Failed to enqueue the message

Cause: The message could not be enqueued.

Action: Contact your Syndication Server administrator.

SYND-730, Cannot open the ZIP file specified by the given name

Cause: The registration (subscriber and content provider) package cannot be found by the given URL.

Action: Correct the URL; check the permission of the package in your local file system.

SYND-731, I/O Exception during the registration process

Cause: I/O error.

Action: Check the local environment or contact Oracle Customer Support Services.

SYND-732, Failed to get the descriptor file for the registration package

Cause: The MANIFEST file in the registration package is not pointing to the descriptor file.

Action: Check the registration package.

SYND-733, XMLException during the registration process

Cause: Failed to validate the descriptor file against the DTD. For example, if the registration process is for a subscriber, the descriptor file should conform to the Subscriber registration DTD.

Action: Check the descriptor file and its corresponding DTD.

SYND-734, Failed to load the specified resource (such as the .xsd file)

Cause: Cannot load the source from the environment (path, env, and so forth).

Action: Check the environment setting.

SYND-735, Gateway entry not found

Cause: Invalid gateway ID or the gateway entry has been removed.

Action: Verify the correctness of the gateway ID. If the problem persists, contact your Syndication Server administrator.

SYND-736, Gateway entry already exists

Cause: Internal administrator error. You tried to add a duplicate gateway entry.

Action: Stop the operation and use the existing one instead.

SYND-741, Failed to set cookie acceptance

Cause: The Oracle HTTP client library does not have the option of accepting an incoming cookie.

Action: Upgrade the Oracle HTTP client library.

SYND-742, Failed to build a valid SSO URL during the handling of redirection

Cause: The Syndication Server client library could not get the login server URL from the HTTP response of the login server.

Action: Check the version of the login server; the Syndication Server client supports only the Oracle9*i*AS V2 login server.

SYND-743, Internal Error: The Syndication Server client library failed to parse the response sent from the login server while it tried to authenticate the current connected Syndication Server client with the Syndication Server SSO module

Cause: The Syndication Server client library could not get the values of the required fields from the login page of the login server.

Action: Check the login page of the login server, the Syndication Server client supports only the Oracle9*i*AS V2 login server.

SYND-744, Oracle cookie module does not exist in the HTTP connection

Cause: The Oracle HTTP client library does not include the required cookie module.

Action: Update the Oracle HTTP client library.

SYND-745, Failed to enable the flag that indicates the current Syndication Server client is already successfully logged onto the Syndication Server system

Cause: Failed to set the cookie indicating that the current Syndication Server client is already authenticated.

Action: Update the Oracle HTTP client library.

SYND-750, Failed to send out email

Cause: The e-mail could not be sent.

Action: Check your e-mail server setting.

SYND-760, Client session expired

Cause: The Syndication Server client session expired.

Action: The Syndication Server client must log in again to refresh the session.

SYND-761, Requested handler not available

Cause: The Syndication Server client does not have the required handler.

Action: Check the request.

SYND-770, Failed to open HTTPS certifcates

Cause: The Syndication Server library could not find the required HTTPS certificates.

Action: Check your HTTPS cerficate path setting in the Syndication Server properties table.

SYND-771, Failed to deliver Ice-item-ref data

Cause: The Syndication Server library could not retrieve the content of ice-item-ref.

Action: Check the definition of the specified ice-item-ref.

SYND-772, Invalid proxy setting

Cause: Incorrect proxy setting.

Action: Check your proxy setting in the Syndication Server properties table.

E

Extensibile Features

This appendix describes some extensible features of Oracle9*i*AS Syndication Server.

E.1 Adding Push Delivery Mechanisms

The mechanism used to extend the scope of *push* delivery is built into Syndication Server as an extensible framework for plugging in new push delivery channels. The push mechanism that is prescribed by ICE is through an HTTP POST request to an HTTP listener that is hosted by the subscriber application. You can extend the scope of push delivery by allowing Syndication Server to push content through other protocols, this time to handle the delivery path, thus allowing for easy extensibility.

Whenever the scheduler informs Syndication Server of an update to be pushed to the subscriber application, a content provider operation module is invoked with the content package as the service request. Currently, Syndication Server extensibility maps one delivery protocol to one content provider operation module. In the case of HTTP delivery, the content provider operation module is an HTTP content provider operation module that takes the content package and makes a POST request to the HTTP listener on the subscriber application's side.

Being extensible to allow different channel protocols for push delivery has an implication on the selection of the content provider operation module at runtime. The mechanism is one in which the subscriber application URL is used to determine the channel protocol and therefore the content provider operation module that is needed. For example, the subscriber application URL `http://www.oracle.com/syndication/Client` triggers the invocation of the push service associated with HTTP, and the subscriber application URL `mailto:oss@oss.com` triggers the invocation of the push service associated with the mailto URL.

E.1.1 Writing New Push Content Provider Operation Modules

The best way to start writing new push content provider operation modules is to copy the HTTP push operation module package (SyndicationPushPackage, located in the `/etc/services` directory on UNIX systems or `\etc\services` directory on Windows systems) to a new name, and then unzip them into some directory so you can modify them. Before modifying these files, read Section E.1.2 and Section E.1.3 for instructions on rules to follow during modifications.

E.1.2 Using Syndication Server Naming Conventions for Naming the Content Provider Operation Module

There is a heuristic that Syndication Server follows in retrieving push content provider operation modules. The identifier for the service must have the following format:

```
"urn:com.push:<push-protocol>.oss"
```

Where `<push-protocol>` is the channel protocol with which to push the content. The heuristic followed maps HTTP for `http` and SMTP to `mailto` and so forth.

E.1.3 Push Content Provider Operation Module Interface: SyndicationPushPackage

Each of these content provider operation modules must conform to some interface, so that it allows Syndication Server to always be able to pass in the content package as a request and receive an acknowledgment back as a response.

E.1.3.1 Input Schema

The input schema envelopes the ice-package element with an ice-push-package element containing an attribute specifying the push URL. The service can forward this URL to the protocol handler to perform the protocol-specific actions for this URL.

E.1.3.2 Output Schema

The output schema models the ice-code element returning either an OK message or the error encapsulated in a comprehensive structure.

E.1.4 Registering the Service

After you have developed and registered your new push service, it is ready for use. The next subscription that contains a subscriber application URL with a protocol

matching the registered service will be able to start using your new service during push delivery.

F

Deploying Oracle9*i*AS Syndication Server

This appendix describes the requirements for Oracle9*i*AS Syndication Server and how to manually deploy Syndication Server into the J2EE (OC4J) environment. This information is provided only if you must manually deploy Syndication Server.

Typically, during an installation of Oracle9*i*AS Portal, Syndication Server is installed and deployed. You need to follow these steps for manual deployment of Syndication Server only in the event that Syndication Server is not already deployed.

Requirements

The following requirements are assumed to be installed and running:

- The Syndication Server database schema has already been installed with the Oracle9*i*AS repository.
- The Oracle9*i*AS instance is up and running (with OC4J).
- JDK version 1.3.1 is already installed and available in the path.
- Check the JDK version by running the following command:

  ```
  java -version
  ```

- The `syndserver.ear` file is located in the following directory:

  <$ORACLE_IASHOME>`/syndication/j2ee/`

Deploying Syndication Server to the J2EE (OC4J) Environment

The following instructions deploy Syndication Server to the J2EE (OC4J) environment:

1. Run the following Java command to deploy Syndication Server to the J2EE (OC4J) environment:

```
java -jar admin.jar ormi://<your-ias-host-name>:<rmi-port> adminuser
adminpwd -deploy -file $IAS_HOME/syndication/j2ee/syndserver.ear
-deploymentName <application-deployment-name>
```

Sample output appears as follows:

```
Auto-unpacking $IAS_HOME/j2ee/home/applications/_syndserver.ear... done.
Auto-unpacking $IAS_HOME/j2ee/home/applications/_syndserver/syndserver.war... done.
Auto-deploying YourAppName (New server version detected)...
```

2. Run the following Java command to bind Syndication Server with the Web application for OC4J:

```
java -jar admin.jar ormi://<your-ias-host-name>:<rmi-port> adminuser adminpwd -bindWebApp
<application-deployment-name> syndserver <web-site-name>/syndserver
```

Note: The name *<application-deployment-name>* specified in Step 1 and Step 2 must be the same name.

By default *<web-site-name>* is the Oracle default `default-web-site`.

When you invoke the following URL, the Syndication Server information appears in a browser window to indicate that Syndication Server is up and running:

`http://<your-host-name>:<your-port-number>/syndserver/OSS`

Glossary

affiliated networks

Members of a group that have formed partnerships to establish ways to share information from a variety of data sources for reuse, such as aggregating and redistributing it.

business-to-business value chain

A business that provides a product to a partner who adds value to it and distributes it to one or more partners, and so forth. Each partner may add value to the product by combining it with other products to form a unique end product. In the context of syndication, the product is content that may be combined with content from other content providers. Syndicators form business partnerships with content providers who aggregate the content and provide catalog offerings from which subscriber applications can choose and establish subscriptions to receive the selected content.

catalog

Groups of content offerings. A subscriber application obtains a catalog offering from a syndicator, and uses the offers within the catalog to initiate the ICE subscription protocol.

collection

A set of items generated over time by receipt of a package sequence. For example, on a subscriber application's site, a collection would be all the different items acquired during the subscription's duration.

content provider

A source of content for subscriber applications. Oracle9*i*AS Syndication Server supports three types of content providers: file, database, and Web, and is also extensible so it can support additional types of content providers.

content provider operation module

A component within the Internet computing model that delivers a specialized value-added function, such as content from a specific data resource not supported by Oracle9*i*AS Syndication Server.

content syndicator

See **syndicator**.

Distinguished Name (DN)

The unique name of a directory entry in Oracle Internet Directory (OID). It includes all the individual names of the parent entries back to the root. The Distinguished Name tells you exactly where the entry resides in the directory's hierarchy. This hierarchy is represented by a directory information tree (DIT).

Document Type Definition (DTD)

An XML document used to validate other XML documents. Oracle9*i*AS Syndication Server uses an ICE agent to take a content syndication or a content subscription request in the form of an XML document, which is validated against a predetermined ICE protocol Document Type Definition (DTD).

ICE

Information and Content Exchange.

ICE payload

See **payload**.

ICE/HTTP

The specific binding of the ICE protocol to the HTTP protocol.

item

A single delivery instance of an arbitrary data type.

message

The abstract concept of an atomic unit of communication.

minimal subscriber application

A subscriber application ICE implementation that has no persistent server component and therefore cannot receive syndicator-initiated request messages.

negotiation

A subscriber application ICE implementation may negotiate with to arrive at mutually agreeable delivery methods and schedules.

operation module provider

A business partner or application developer who provides and manages the content of an operation module. Typically, it is the owner of some data resource or process, such as, the owner of a currency exchange rate Web site. Also, it is someone who provides content for an operation module.

package

A single delivery instance of a group of items. For example, a single issue of a parts manual or a single set of headlines. A package is the atomic unit of information distribution in ICE.

package sequence

An ordered series of packages delivered over time.

payload

A protocol structure that includes a set of logical ICE operations delivered at discrete intervals. A payload is a single instance of an XML document formatted according to the protocol definitions contained in the ICE 1.1 specification.

provider

See **content provider**.

pull

The retrieval of content by a subscriber application who contacts a syndicator for new or updated content provided by content providers.

push

The automatic delivery of content by sending it to subscriber applications when content changes or according to some negotiated schedule.

receiver

Generic term referring to the target of an ICE payload.

request

A message asking for the performance of an operation. Requests in ICE are messages carried by payloads.

requester

Generic term referring to the initiator of an ICE payload request.

responder

Generic term referring to the recipient of an ICE payload request.

response

A message containing the results of an operation. Responses in ICE are messages carried by payloads.

sender

Generic term referring to the originator of an ICE payload.

subscriber application or subscriber

In the world of content syndication, one of the two parties in an ICE relationship, who receives information and content from the other party, a syndicator. The subscriber application uses ICE to obtain information and content from the syndicator.

subscription

An agreement to deliver a package sequence from a syndicator to a subscriber application. There may be many independent subscriptions between a syndicator and a subscriber application.

subscription element

The identifier for a logical concept represented by a specific item, or group of items, within a collection. The subscription element may have many versions over time, and thus may have been represented by different items. For example, a company logo is a single subscription element, that can be updated over time. Every subscription element has a unique subscription element ID assigned by the syndicator.

subscription offer

A proposed set of parameters for a particular subscription.

syndicator

In the world of content syndication, one of the two parties in an ICE relationship, who sends information and content to the other party, a subscriber application. The syndicator uses ICE to send information and content to the subscriber application. The role of content syndicator is carried out by Oracle9*i*AS Syndication Server.

unsolicited message

A protocol mechanism used in ICE to provide a way for a syndicator to initiate communication to a minimal subscriber application.

Index