Connection Server Installation Guide

10*g* Release 3 (10.1.3.3.0)

March 2007

ORACLE®

Connection Server Installation Guide, 10*g* Release 3 (10.1.3.3.0)

Contributing Author: Will Harris

Contributor: Greg Brown

Contents

CHAPTER 1

Introduction

Connection Server provides the ability to pull content from multiple repositories and feed it into Content Server. It also provides the ability to route content stored in Content Server to other repositories as well as to partner sites or consumers and can perform functions such as routing catalog content to partner extranets so that partners automatically receive the content when they access the extranet.

Connection Server is a digital distribution/Syndication engine. It moves digital assets from one point to a another point or to multiple points. Connection Server monitors a collection of assets defined as an 'offer' and it determines if an asset has been added, modified, or removed from the collection. Subscribers/Delivery points are designed to receive the offer with an associated delivery rule defined by a subscription.

Connection Server handles the distribution of dynamic offers (collection of assets). A dynamic offer is where content is being added, removed or changed within the collection. The value of Connection Server increases when you have many points that need to receive content feeds. Connection Server enforces the rules that users define for the Subscriptions.

Connection server is a Java based solution that allows content providers to control distribution and user access and provides solutions for integrating disconnected Islands of Information, deployment of content to Server Farms, and distribution of content to subscribers

Connection Server utilizes industry standard ICE (Information & Content Exchange) protocol, supports SSL (Secure Socket Layer) 40-bit and 128-bit security, provides subscription based content delivery, and allows content metrics to measure page views, unique visitors, and track activity by content offer or by subscribers.

In this section:

- Installation requirements
- Deployment overview
- Default port assignments
- Firewall considerations
- Windows FTP Server configuration
- New features and changes
- Conventions used in this guide
- About the Help system

Installation requirements

Operating system, database, and JVM requirements must be met before installing Connection Server.

Operating system

Connection Server is supported on these operating systems:

- Windows: 2000, XP, and 2003.
- UNIX-based: Solaris and Linux.

Database

Connection Server requires one of these databases:

- Microsoft SQL Server 2000 with Service Pack 2 (SP2) or greater.
- IBM DB2 8.0.1 or greater.
- Oracle 9 or 10 database.

Java Virtual Machine

Connection Server requires that J2SE 1.4 SDK be installed on your machine. The latest patch release is recommended (J2SE v 1.4.2_10 or greater).

You can download a Java Virtual Machine from the http://java.sun.com/ web site. Download the version for your operating system and install it according to the installation instructions provided.

Important: Do not use Microsoft Jview or the JRE.

Notes

- For Windows platforms, if you will be distributing English-only content, download the US English version. If you need support for international character sets, download the Internationalized version.
- Make sure the JVM location (the fully-qualified path to its installed directory) is included in your computer's PATH environment variable.
- For Windows platforms, refer to the Windows help system for information on editing the PATH variable. Also, set the JAVA_HOME environment variable to the JVM installation directory.

Setting the Max Heap Size

The Max Heap Size is the maximum amount of physical memory allocated to the Java Virtual Machine (JVM) running the Connection Server. The default configuration is 128 MB for the Connection Server. For some production implementations this setting is too low. A low memory setting will result in slower performance and ultimately out-of-memory messages in the log files.

If you need to increase the memory allocation, it is recommended that the default be changed to 256 MB. Changing the Max Heap Size setting must be done in two places:

- **cns.wrapper.conf file** — Change the value for the variable wrapper.java.maxmemory. This file is located in the Connection Server install root /conf/ directory.

- **cns.bat** (or **cns.sh**) — Change the value in the command line argument from -mx128m to -mx256m. This file is located in the Connection Server install root directory.

It is also recommended that the Max Heap Size for Subscription Client be changed to128mb. The default is 64mb. Changing the Max Heap Size setting must be done in two places for the Subscription Client.

- **sub.wrapper.conf** — Change the value for the variable wrapper.java.maxmemory. This file is located in the Subscription Client install root /conf/ directory.

- **launch-program.bat** (or **launch-program.sh**) — Change the value in the command line argument from -mx64m to -mx128m. This file is located in the Subscription Client install root directory.

  For example:

  C:/j2sdk1.4.2_03/bin/java -mx128m -Djava.security.auth.login.config="./conf/cns_client_jaas.config" bootstrap.SubscriberStart $

Deployment overview

This is a high-level overview for deploying Connection Server in a multi-site delivery configuration with Content Server.

To set up the deployment environment

- Install and configure Connection Server to get files (an offer) from your Content Server instance.

- Install a Subscription Client (or FTP client) on each machine that needs to have these files copied to it. Each Subscription client subscribes to the set of files (the offer) that need to be deployed to that machine.

- The Connection Server administrator chooses when and how the files get sent to each subscriber machine.

To set up the Connection Server

- Create a Delivery Rule that describes when you want content to be deployed to the subscribers.

- Create an Offer: The Content Source Type should be **File System - Directory** and set the Advanced Delivery Options to **Synchronized**. Choose **Next** and pick the directory to deploy.

- Create a User for each subscriber machine. Choose the Delivery Method, and assign the role **Subscriber** to the user.

- Subscribe each User to the Offer.

To set up a subscriber

- Add the Connection Server as a Content Provider.

The Connection Server Password is the one entered when the User was created, and the Client UUID is automatically generated. It can be viewed on the Connection Server by logging in with the User ID and Password entered when the user was created.

Default port assignments

The default port numbers used by the Connection Server are:

Port Number	Uses	Component
8885	push server for SSL	Subscription Client
8886	push server	Subscription Client
8887	administration server	Subscription Client
8880	master server	Connection Server
8889	administration server (this is the Publish Server URL port number)	Connection Server
8890	ICE server	Connection Server
8891	file server	Connection Server
8893	ICE server for SSL	Connection Server
8894	file server for SSL	Connection Server

Firewall considerations

If your company intranet is protected by a firewall, one or more ports may need to be opened through the firewall to allow Subscription Client and Connection Server to communicate.

Connection server components may be placed outside or behind a firewall, depending on the requirements at your site. For example, you might place Content Server outside the firewall, but the ICE Server and the Administration Server/Package Generator combination behind the firewall.

Subscription Client communicates with both the ICE Server and Content Server. Under normal distribution operations, the Administration Server communicates only internally, not across the Internet, so this component can reside behind the firewall and its port can remain closed. If your site requires that the Administration Server be accessed from the Internet, then it's port must be opened, as well. With a configuration that includes failover servers, ports to failover servers that reside behind a firewall must be opened, as well.

If you use **push** subscriptions, and the Subscription Clients receiving these subscriptions reside behind a firewall, the Subscription Client port must be opened through the firewall to receive the Connection Server ICE messages.

Windows FTP Server configuration

Connection Server supports Windows FTP Servers only in the UNIX Directory Listing Style, not with MS-DOS directory style.

1. Open IIS Manager and select **FTP Sites**.

2. Click **Action > Properties** from the main menu.

3. Select the **Home Directory** tab.

4. Directory Listing Style: enable the **UNIX** option.

5. FTP Site Directory: enable the **Write** option.

6. Click **OK** and select the related child-nodes. The Default FTP Site must be set to writable.

7. Click **OK** and restart the FTP server.

New features and changes

These new features and changes are included in this release:

- Database Compaction — Allows the server to shrink the tables in the database that are used to track package updates (removes older redundant information from the database).

- Database driver upgrade for Oracle (we support 9i and above).

- Subscription Client can now be run as a service.

Conventions used in this guide

The User Guide contains the following conventions:

Convention	Definition
Bold	Indicates an item that you select in the interface, such as a button or menu, in order to perform a specific task: Click **OK** to confirm the deletion.
>	Indicates a menu choice. For example, "Choose **File > Open**" means "Click the **File** menu, and then click **Open**."
`Code`	Indicates actual code.

Notes

- The prefix CNS_HOME/ refers to the local directory in which Connection Server is installed. For example, if you installed on a Windows platform and accepted the installer's default target directory, this directory would be something like: C:\Program Files\Stellent\Connection Server. If you installed on a UNIX platform, this might be: /stellent/ (or another arbitrarily named directory).

- The prefix Content_Server_path/ refers to the local directory in which Content Server is installed (not Connection Server). If you installed on a Windows platform and accepted the installation default directory, this directory would be something like: C:\Program Files\Stellent\Connection Server. If you installed on a Solaris platform, this might be: /stellent (or another named directory).

- The prefix Subscription_Client_path/ refers to the Subscription Client installation directory.

- The "Connection Server configuration file" refers to either *cns.oracle.config* or *cns.sqlserver.config* depending on your database.

- Forward slashes (/) are often used to separate the directory levels in a path name. Windows users should substitute backslashes (\) where appropriate.

About the Help system

Online Help is accessible from the Connection Server administration interface. After installation, you can launch the administration interface in your web browser (example: http://testserver7:8889).

CHAPTER 2

Installation steps

This section describes how to install and configure Connection Server and Subscription Client. The Connection Server is the server side of the system and either a Subscription Client or an FTP server serves as the client side of the system. The Subscription Client is generally installed on every machine that hosts a web server (unless you choose to use an FTP server).

You will need to enter the provided license key as the Connection Server UUID when you first start up the Connection Server.

This section provides information on the installation of Connection Server and includes the following topics:

- Installing Connection Server
- Configure General Settings
- Installing the Subscription Client
- Configuring the Subscription Client
- Uninstalling Connection Server and Subscription Client

Installing Connection Server

Connection Server may be installed on the same machine as Content Server. However, we recommend that the Connection Server be installed on a dedicated server.

To install the Connection Server

1. Insert the DVD or unbundle the distribution ZIP file.

2. Navigate to the \win32\ directory and launch **Setup.exe** (for UNIX, run the /unix/install.sh executable).

3. When the Welcome screen displays, click **Next**.

4. Click **Yes** to accept the license agreement.

5. If this is a new installation, select **Install a new Connection Server**.

   If you are updating Connection Server, select **Update Connection Server** and follow the prompts, selecting the directory where your existing Connection Server is installed."

6. Click **Next**.

7. Accept the default directory (recommended) or enter a destination folder.

8. Click **Next**.

9. Enter the path to the Java Virtual Machine (J2SE v 1.4.2_10 or greater). See "Java Virtual Machine" on page 2 for more information.

10. Click **Next**.

11. Review the settings before copying the files and click **Next**.

12. Click **Finish**.

Setting up the database

This section explains how to configure or upgrade the Connection Server for each of the supported database management systems.

Configure Oracle driver

If you have not yet created an Oracle database for use with the Connection Server, create one first and then run the SQL script syn_server.oracle8.sql to create tables. The script can be found in your Connection Server\sql directory. Refer to your Oracle documentation for information about running SQL scripts.

Note: The database tables owned by the login associated with the Connection Server must be created in their own dedicated tablespace.

If your database is not set up properly, the following error messages will be returned:

```
database driver could not be loaded
failed to open the database
```

Configure Microsoft SQL Server driver

In SQL Server, create a database. In the new database, use SQL Query Analyzer to load and execute the syn_server.mssql2000.sql creation script located in your Connection Server \sql\ subdirectory. This script will create the tables within the database. Refer to your Microsoft SQL Server 2000 documentation for information about running SQL scripts.

If your database is not set up properly, the following error messages will be returned:

```
database driver could not be loaded
failed to open the database
```

Configure DB2 driver

DB2 driver is not included in the Connection Server CD or distribution zip file. You must copy syn_server.db2.sql from CNS_HOME/sql/ to a directory on your DB2 host. Then, copy the db2jcc.jar or db2jcc.zip file from the DB2 installation directory sqllib/java/ to the CNS_HOME/lib/ directory. And finally, create database and tables in DB2 (using the DB2 Command Window).

For example:

```
db2 CREATE DB <db anme> ALIAS <db alias>
db2 CONNECT TO <db alias> USER <user> USING <password>
db2 -tvf syn_server.db2.sql
```

Connect the Connection Server to your database

This section describes how to configure your Connection Server installation for the appropriate database.

To connect to Oracle

1. Using a text-only editor, open the cns.oracle.config file located in the directory where your Connection Server is installed. This is the sample configuration file for Oracle (see below).

```
<database type="oracle">
<driver jdbcURL="jdbc:oracle:thin:@SERVER:1521:NAME"
driver="oracle.jdbc.driver.OracleDriver"/>
<user username="USER" password="PASSWORD"/>
</database>
```

2. Replace SERVER with the name of your Oracle server (e.g., "10.10.1.248" or "server7").

3. The default port is 1521. Change this, if necessary.

4. Replace NAME with the name you have given the database on the server (e.g., "sspu").

5. Replace USER and PASSWORD with the user name and password used to access the database.

For example:

```
<database type="oracle">
<driver jdbcURL="jdbc:oracle:thin:@server7:1521:sspu"
driver="oracle.jdbc.driver.OracleDriver"/>
<user username="administrator" password="administrator"/>
</database>
```

6. Start Connection Server by executing the cns.oracle.bat file.

Note: The default administrator user name and password is administrator / administrator.

To connect to DB2

1. Using a text-only editor, open the cns.db2.config file located in the directory where your Connection Server is installed. This is the sample configuration file for DB2 (see below).

```
<database type="db2">
<driver jdbcURL="jdbc:db2://SERVER:50000/NAME"
driver="com.ibm.db2.jcc.DB2Driver"/>
<user username="USER" password="PASSWORD"/>
</database>
```

2. Replace SERVER with the name of your DB2 server (e.g., "10.10.1.248" or "server7").

3. The default port is 50000. Change this, if necessary.

4. Replace NAME with the name you have given the database on the server (e.g., "sspu").

5. Replace USER and PASSWORD with the user name and password used to access the database.

For example:

```
<database type="db2">
<driver jdbcURL="jdbc:db2://server7:50000/sspu"
driver="com.ibm.db2.jcc.DB2Driver"/>
<user username="administrator" password="administrator"/>
</database>
```

6. Save and close the file.

7. Start Connection Server by executing the cns.db2.bat file.

Note: The default administrator user name and password is administrator / administrator.

To connect to MS SQL Server

1. Using a text-only editor, open the cns.sqlserver.config file located in the directory where your Connection Server is installed. This is the sample configuration file for SQL Server (see below).

```
<database type="mssql">
<driver jdbcURL="jdbc:microsoft:sqlserver://SERVER:1433;
DatabaseName=NAME; SelectMethod=cursor"
driver="com.microsoft.jdbc.sqlserver.SQLServerDriver"/>
<user username="USER" password="PASSWOORD"/>
```

2. Replace SERVER with the name of your SQL server (e.g., "10.10.1.248" or "server7").

3. The default port is 1433. Change this, if necessary.

4. Replace NAME with the name you have given the database on the server (e.g., "sspu").

    Refer to your Microsoft (SQL Server) JDBC driver's documentation for more information on using the DatabaseName property in the connection string.

5. For the SelectMethod property, use the "cursor" select method.

6. Append the driver path.

    For example:

    ```
    com.microsoft.jdbc.sqlserver.SQLServerDriver
    ```

7. Replace USER and PASSWORD with the user name and password used to access the database.

    For example:

```
<database type="mssql">
<driver jdbcURL="jdbc:microsoft:sqlserver://server7:1433;
DatabaseName=sspu; SelectMethod=cursor"
driver="com.microsoft.jdbc.sqlserver.SQLServerDriver"/>
<user username="administrator" password="administrator"/>
</database>
```

8. Start Connection Server by executing the cns.sqlserver.bat file.

Note: The default administrator user name and password is administrator / administrator.

Installing as a Windows service

You can install the Connection Server as a service on Windows. This will allow you to keep the Connection Server running after you have logged out, and it will inhibit unauthorized users from tampering with the Connection Server's operation.

To install as a Windows service

1. Make sure that you are logged into the machine with administrator rights.

2. Using a text-only editor, open the cns.wrapper.config file located in your Connection Server \conf\ subdirectory.

3. Edit the configuration file (make sure that you have the correct Connection Server configuration file specified in cns.wrapper.config):

- For MS SqlServer (default)

  ```
  wrapper.app.parameter.1=cns.sqlserver.config
  ```

- For Oracle

  ```
  wrapper.app.parameter.1=cns.oracle.config
  ```

- For DB2

  ```
  wrapper.app.parameter.1=cns.db2.config
  ```

4. Open a Command Prompt window:

   - In the Start menu, click **Run**.

   - In the Run dialog box, type **cmd**.

   - Click **OK** to open the Command Prompt window.

5. Change to the directory where you installed the Connection Server.

6. Type **cns.service-install.bat**.

This process installs the service; it does not start it. You can start the service from the Control Panel Services applet or by rebooting.

Note: If you choose to remove the service, enter the following on a command line, from the same Connection Server installation directory: **cns.service-uninstall.bat**.

Configure General Settings

Follow these steps to configure the general settings:

1. Start Connection Server by running the startup file for your configuration *cns.oracle.bat*, *cns.sqlserver.bat or cns.db2.bat* depending on your database.

2. Launch the administration interface in your web browser
   (example: http://testserver7:8889).

3. Enter your administrator user name and password to access the administration interface, and click **OK**.

4. On the administration interface, click **General Settings**.

5. Enter the Connection Server Identity information.

   Connection Server Name (required): The name of this Connection Server—usually your company's name.

   Connection Server UUID (required): The Universal Unique Identifier (UUID) for your Connection Server. You must provide a valid UUID the first time you start the Connection Server. Your Connection Server's UUID is displayed on the Subscription Client Deployment Server, on the Change Profile page.

   Description: Information about your site—usually a description of your business or offers. This description is written to the configuration file of Subscription Client.

   Information URL: The URL of a web page that gives subscribing companies more information about you or your service.

6. When done, click **Save**.

7. On the settings confirmation screen click **OK**.

Important Configuration Notes

Please follow these configuration instructions:

- The description for Push Immediate is inaccurate; a delivery frequency must be specified for the delivery to function correctly. This option is on the "Delivery Rule" edit page.

- It is recommended that Needs Confirmation be selected for all subscriptions. This option is in the "Delivery Rule" section of the subscription details.

- It is recommended that Send Removals be used on all Filewatcher offers. This option is on the second Filewatcher page at the bottom, under the update schedule.

- It is recommended that the Purge Content Items field be left blank (thus disabling it) due to the limitations of database purging. This option is in the "Database Purge" section.

Administration interface

Most day-to-day usage and ordinary configuration tasks can be done using the browser-based Connection Server administration interface. This section provides information on the Connection Server administration page.

Menu	Description
Home	Returns you to the main Connection Server Menu page.
Offers	Offers are packages of content.
Subscriptions	Subscriptions link Offers, Users and delivery information.
Delivery Rules	Delivery Rules define the dates and times at which various actions, such as Subscription delivery, are executed.
Content Directories	Content Directories are web catalogs which display offers within categories.
Users	Users are assigned one or more roles, such as Subscriber or Administrator.
Roles	Roles are sets of permissions assigned to User Groups and Users.
User Groups	User Groups are groups available to Users and own Roles and Offers.
General Settings	General Settings set system options.
Help	Provides online help information: - The Connection Server pages have help outlining what you can do on that page. Click the large help icon next to the title of the page to view page level help. - Most Connection Server pages are divided into sections. To get help for a section and the individual fields in it, click the help icon in the section header.
About	The About Connection Server page provides this information: - Connection Server version. - Connection Server build date. - Maximum subscribers allowed. - Java Virtual Machine version.

Menu	Description
Support	Links to http://www.stellent.com
Shut Down	Shut down or restart.

Installing the Subscription Client

Subscription Client is the client-side component of the Connection Server. The Subscription Client is generally installed on every machine that hosts a web server (unless you choose to use an FTP server). This is the destination for publishing.

To install the Subscription Client

1. Insert the DVD or unbundle the distribution Zip file.

2. Navigate to your Subscription Client `\win32\` subdirectory and launch **Setup.exe** (for UNIX, run the /unix/install.sh executable).

3. When the InstallShield Wizard displays, click **Next**.

4. Click **Yes** to accept the license agreement.

5. If this is a new installation, select **Install a new Subscription Client**.

   If you are updating Subscription Client, select **Update Subscription Client** and follow the prompts, selecting the directory where your existing Subscription Client is installed."

6. Click **Next**.

7. Accept the default directory (recommended) or enter a destination folder.

8. Click **Next**.

9. Enter the path to the Java Virtual Machine (J2SE v 1.4.2_10 or greater). See "Java Virtual Machine" on page 2 for more information.

10. Click **Next**.

11. Review the settings before copying the files and click **Next**.

12. Click **Finish**.

Configuring the Subscription Client

Follow these steps to configure the Subscription Client to communicate with the Connection Server.

To configure the Subscription Client

1. Start the Subscription Client by running the startup file for your configuration.

   The batch file sub_agent.bat (for Windows) and the shell script sub_agent.sh (for UNIX) are located in the directory where the Subscription Client is installed.

2. Access the Subscription Client Administration interface (e.g., http://localserver:8887) and log in as an administrator (the default user name and password is administrator / administrator).

3. On the Subscription Client Administration interface, click **Content Providers**.

4. Click **Add Content Provider**.

5. Enter these values:

- **Connection Server URL:** The URL for your Connection Server instance (e.g., http://mainserver).

- **Subscription Client UUID:** The UUID can be found on the Connection Server administration interface by clicking **Users**.

- **Connection Server Password:** Enter your Connection Server password. This password can be found on the Connection Server administration interface by clicking **Users** and then clicking the destination name (for example, "Server7").

- **Local Directory:** Enter a directory (or browse to a location) where the delivered content will reside.

6. Click **Save**.

Uninstalling Connection Server and Subscription Client

If you no longer want to use Connection Server or Subscription Client, you can uninstall them by following the steps below.

To uninstall Connection Server

1. Insert the Connection Server DVD.

2. Navigate to the Connection Server `\win32\` directory.

3. Double click **Setup.exe.**

    Note: For UNIX, run the /unix/install.sh executable.

4. Choose the **Remove a Connection Server** option.

5. Follow the instructions on the screen to uninstall the Connection Server.

To uninstall Subscription Client

1. Insert the Connection Server DVD.

2. Navigate to your Subscription Client `\win32\` subdirectory.

3. Double click **Setup.exe.**

    Note: For UNIX, run the /unix/install.sh executable.

4. Choose the **Remove a Subscription Client** option.

5. Follow the instructions on the screen to uninstall Subscription Client.

CHAPTER 3

Subscription Client administration

Subscription Client is the client-side component of the Connection Server. The Subscription Client is generally installed on every machine that hosts a web server (unless you choose to use an FTP server).

In this section:

- Subscription Client administration interface
- Subscription Client general settings
- Subscription Client configuration file

Subscription Client administration interface

Start the Subscription Client by running the startup file for your configuration. The batch file sub_agent.bat (for Windows) and the shell script sub_agent.sh (for UNIX) are located in the directory where the Subscription Client is installed.

Access the Subscription Client Administration interface (e.g., http://localserver:8887) and log in as an administrator (the default user name and password is administrator / administrator).

Controlling the browser display

Important: The software must be shut down for any edits to the configuration file to take effect. If you edit while the software is running, a warning dialog will display the next time you shut down, stating that the changes you made to the file will be overwritten.

You can control whether the Status window or the Subscription Client administration interface displays at startup by changing the Startup Options in the Settings page of the administration interface, or by changing the following <options> elements in the Subscription Client configuration file (siclone.config):

- **run-interactive:** To display the Status Window at startup, set to true; to prevent it from displaying at startup, set to false.
- **start-browser:** To display the administration interface at startup, set to true; to prevent the interface from displaying at startup, set to false

Example:

```
<options run-interactive="true" start-browser="true">
```

Stopping the Subscription Client

When you stop the Subscription Client, you may shut it down completely or you may shut it down and immediately restart it. If you are using the administration interface and Status Window, always close Subscription Client using the Shutdown option in the Subscription Client navigation bar.

Subscription Client general settings

The following settings identify your Subscription Client and determine its operational properties:

- **Identity:** Provides security for the Subscription Client administration interface. You can specify an identity name and password to prevent non-administration users from changing important settings. The identity name and password strings are case-sensitive.

- **Proxy Server:** An intermediate server in your network that receives requests for Internet access. If a proxy server is in use at your site, configure the Subscription Client to use it. See a network administrator at your site for information about proxy servers in your environment.

  For a proxy server, you specify its IP address and port number; if your proxy server is secure, you must provide the port number on which the proxy server listens for SSL authentication.

- **Log Settings:** While the Subscription Client is running, it automatically generates a log file named subscriber.log.

  By default, the Subscription Client uses a rotation method to manage its log file. Additional log files are created when the current one reaches its maximum configured size. The Subscription Client continues to generate log files of a constant size until it has generated the number configured. After the specified number of files is reached, the Subscription Client deletes the oldest log file each time it creates a new one. Thus, the most recent logging information is always retained. Using a rotating scheme controls the amount of disk space used for storing log messages.

  The file size for subscriber.log is 1024 KB (maximum size allowed). The default number of files to rotate is 7; the maximum number is 100.

- **Startup Options:** Determine the components of Subscription Client that display at startup. If you have a Windows version of the Subscription Client, both the browser-based Subscription Client Administrator and the Subscription Client Status Window display on initial startup. If you have a UNIX-based version of Subscription Client, neither element displays on initial startup.

- **Auto Upgrade:** Keeps the Subscription Client application up-to-date with the latest software releases. If this option is enabled, application updates are automatically applied to the Subscription Client. The Subscription Client briefly stops, applies the upgrade, then restarts, with minimal downtime. By default, auto upgrade is enabled.

Subscription Client configuration file

The entries in the Subscription Client configuration file (siclone.config) control the operation of the software. The configuration file also stores entries related to extension and customization of the Subscription Client functions.

The configuration file contains these elements: siclone, options, log, proxy, ssl, extensions, and extension.

siclone element

The siclone element is the root element of the configuration file; it is the beginning and ending statement in the file. The siclone element contains a set of global operating settings for the Subscription Client and has these child elements: options (1), defaults (1), extensions (1), and job (any number, including none). It has the following attributes:

Attribute	Purpose	Default
last-run-product-code	name of the component that created the configuration file	subscriber
last-run-version	release level of the Subscription Client that last ran on the PC	none
last-run-os	operating system on the PC where the Subscription Client last ran	none

options element

The options element contains the Subscription Client's global settings and has these child elements: log (1), proxy (1), and ssl (1). It has the following base attributes:

Attribute	Purpose	Values and Default
run-interactive	controls the display of the Status window at startup	**true** displays the Status window at startup **false** disables display of the Status window at startup Default: true
start-browser	controls starting the web browser at startup (with the Subscription Client main menu displayed)	**true** enables startup of a web browser with the Subscription Client main menu page **false** disables startup of a web browser Default: true
browser-path	fully-qualified path name to the web browser you want to use for the Subscription Client; this setting overrides auto-detection of the default web browser; for example: C:\Program Files\NetscapeCommunicator\netscape.exe or /usr/local/bin/netscape	Default: none
max-retries	maximum number of times to retry a failed download	Default: 3
ui-port	port to be monitored for the HTML administration interface	valid port number Default: 8887
admin-username	name required to pass the Administration module's authentication	Default: administrator
admin-password	password for the admin-username above; used to pass the Administration module's authentication; base-64 encoded	Default: administrator
push-port	port where the Subscription Client listens for non-SSL push-mode messages	valid port number Default: 8886

Attribute	Purpose	Values and Default
push-ssl-port	port where the Subscription Client listens for push-mode messages using SSL	valid port number Default: 8885
download-timeout	specifies in milliseconds the amount of time the Subscription Client will wait for a download to commence before timing out.	Default: 300000
max-retries	specifies maximum number of times the Subscription Client will attempt to download a file	Default: 3
min-requestor-check	specifies the amount of time in milliseconds that the requester will check for any messages	Default: 30000

log element

The log element contains information about Subscription Client logging. The log element has no child elements It has the following attributes:

Attribute	Purpose	Values and Default
filesize	maximum size, in KB, of the log file	Default: 1024 (largest size allowed)
numfiles	maximum number of log files to create for rotation	Default: 7 (may be as large as 100)
filepath	fully qualified name of the default log file. If only the filename is specified, it is written in the directory where the Subscription Client is installed	Default: subscriber.log
default	default priority level of log messages	priority levels are (in order of decreasing severity); **critical**—serious error or crash **error**—an operation has failed **warning**—unusual conditions **verbose**—detailed progress messages **info**—informational messages **debug**—for programmer use only; includes all levels Default: error
logtofile	specifies where the log information is written	**true**: outputs to log file **false**: outputs to console Default: true
overwrite	specifies whether log files will be overwritten in rotation	**true**: overwrites old log files as defined in the rotation (see numfiles attribute) **false**: used single log file (non-rotation) Default: true

Logging by facility

You can set the level of severity for events logged from various component facilities of the Subscription Client. The higher the severity level, the fewer the log messages generated.

For any level, log messages for all levels above it are also generated. For example, if the logging level is set to error, messages at the critical level are also generated.

Here is an example of a log element:

```
<options>
<log default="warning" ice="verbose" replicator="debug"/>
</options>
```

This example sets the default logging level to warning, suppressing messages of lower severity and returning messages of higher severity. Messages from the ICE facility are set at the verbose level, while all events, including debugging information, are logged for the replicator facility.

Note: Do not use debug logging level unless specifically instructed.

These are the facilities for which event logging can be specified:

- **analyzer:** messages from the web server related to analysis of content provider information

- **filter:** messages generated by content filtering

- **httpd:** connection and networking messages

- **ice:** ICE server (ice requests exchanged)

- **replicator:** component that copies web pages to local directories

- **scheduler:** records of times of content distribution

- **SKITL:** messages generated by the Kinecta Transformation Language

- **XMLParser:** XML processing messages

proxy element

The proxy element contains information about using a proxy server, if it exists at your site. The proxy element has no child elements. It has the following attributes:

Attribute	Purpose	Values and Default
proxyset	specifies whether proxy server settings are defined	**true**: proxy server settings are defined
		false: proxy server settings are not defined
		Default: false
host	IP address or host name of the proxy server	valid IP address or host name of the proxy server.
port	port number of the proxy server	Default: null (-1)
ssl-port	port number of the Secure Sockets Layer (SSL) security enabled proxy server port.	valid port number
		Default: null (-1)

Attribute	Purpose	Values and Default
username	user ID for logging into a proxy server that performs authentication	valid user ID for secure proxy server.
password	password associated with the proxyUsername. May be plain text or base-64 encoded; use an @ as the first character of a base-64 encoded password specification	valid password for secure proxy server

ssl element

The proxy element contains information about using Secure Socket Layer. The ssl element has no child elements. It has the following attributes:

Attribute	Purpose	Values and Default
enable	Specifies whether SSL security will be used by the Publishing Utility. Set this attribute to "true" to enable SSL.	**true**: The Publishing Utility will use SSL security to connect to the Administration user interface and the ICE server. **false**: The Publishing Utility can only use non-SSL ports. Default: false
required	Specifies whether only SSL security can be used and prohibits the use of other protocols.	**true**: Only SSL can be used. Non-SSL connection are not supported. **false**: Both SSL and non-SSL connections are supported. Default: false
admin	Specifies whether the Administration user interface is enabled over SSL. Set this attribute to "true" to enable this web-based interface.	**true**: The Administration user interface is enabled. **false**: The Administration user interface is not enabled. Default: false
keystorefilename	The SSL certificate file name.	**yourfilename**: Provide your actual certificate file name.
keystorepass	The SSL Keystore password.	**yourpassword**: Provide your actual password.

Secure Sockets Layer (SSL) certificates are not provided for you. You must create a SSL certificate and edit the configuration file to enable SSL. The configuration file for your instance will be "cns.sqlserver.config", "cns.oracle.config", or "cns.db2.config" (depending on your database). An example ssl element would look like this:

```
<ssl enable="true" required="false" admin="true"
    keystorefilename="yourfilename" keystorepass="yourpassword"/>
```

extensions element

The extensions element contains the set of extension entries that implement additional functionality in the Subscription Client. The extensions element has a single child element: extension (this takes any number, including none).

extension element

The extension element specifies a class that implements "com.kinecta.subscriber.Extension". The extension element has no child elements. It has the following attributes:

Attribute	Purpose	Values
class	Java class name to be invoked for an extension	valid Java class name for this extension
param	information to be passed to the Java class when invoked	any string you wish to pass to the extension

CHAPTER 4

Configuration file properties

The entries in the Connection Server startup configuration file can be edited to control the operational characteristics of the Connection Server. The Connection Server configuration file is *cns.oracle.config*, *cns.sqlserver.config or cns.db2.config* depending on your database.

In this section:

- The root element
- The options element
- The services element
- The database element
- The content-sources element
- The extensions element
- The ldap element
- The j2ee element

The root element

The root element of the configuration file must always be syndicator. These elements can appear as child elements:

- options
- services
- database
- content-sources
- extensions.

The options element

This section covers the options element and the following sub-elements:

- The timeFormat element
- The log element
- The proxy element
- The ssl element

Attributes of the options element

The options element contains global settings that govern how the Connection Server behaves. This element contains the following attributes:

Attribute	Purpose	Values and Default
browser-path	Specifies the path to the web browser you use to administer Connection Server. This is required only if you wish to automatically launch your browser when the Connection Server starts *and* if the Connection Server is unable to find it.	Valid file path. Default: The path to the default browser defined to the operating system.
content-browsing	Specifies whether a web browser connecting to the Content Server can view files listed in directories	**true**: Browsing enabled **false**: Browsing disabled Default: true
custom-item-fields	A comma-delimited list of custom metadata properties that, if present, you would want to send alongside the content item in an ICE package. If the metadata property is not listed, or if this attribute is not present, the metadata will not be sent—despite the fact that it was associated with a content item by its content source.	Your list can contain a maximum of 26 properties.
default-pull-delivery-rule-name	This attribute define the names of the default pull delivery rule. The default delivery rules have special limitations and behaviors: for example, they can not be deleted or renamed. The pull/push attributes allow you to change which named delivery rules have these special conditions.	Default: Default Delivery Rule
default-push-delivery-rule-name	This attribute define the names of the default push delivery rule. The default delivery rules have special limitations and behaviors: for example, they can not be deleted or renamed. The pull/push attributes allow you to change which named delivery rules have these special conditions.	Defaults: Default Push Delivery Rule
download-base	Specifies the local directory under which all locally served content is stored	Valid file path

Attribute	Purpose	Values and Default
filewatcher-checksum	Specifies the calculation of checksums on files to determine if a file has changed; replaces the use of the timestamp to determine changes to files. If very large files are being distributed, the checksum calculation may cause a negative performance impact.	**true**: Checksum calculation enabled **false**: Checksum calculation disabled. Default: false
hostname	Allows you to override the default hostname that is automatically determined at startup	Valid hostname. *Default:* Determined at startup To find the current hostname, check this line at Connection Server start-up: "Starting Connection Server on <hostname>"
ice-cache-silence-time-out	If this attribute is enabled (and caching is enabled), the Connection Server considers that an error has occurred if the cache does not detect any new item added to, or removed from, the database within the specified time. The Connection Server then refuses any connection and returns an ICE code 503 error message until new activity is detected. It also logs this error in the log file.	**0**: disabled **>0**: specified time in milliseconds Default: 0
ice-cache-size	Specifies the maximum number of ICE items that the Connection Server's cache can contain	*Default:* **5000**
ice-cache-update-interval	Specifies in milliseconds how often the Connection Server queries the database to update its ICE items cache	*Default:* **10000** (10 seconds)
ice-failover-adjustment	If this attribute is enabled when a Subscription Client connects to an ICE server running as a backup server, or if a Subscription Client was previously connected to an ICE server running as a backup server, the Connection Server rolls back the subscription state by the number of milliseconds specified. As a result, redundant adds or removes may be sent.	**0**: disabled **>0**: number of milliseconds to roll back Default: 0

Attribute	Purpose	Values and Default
intra-distributed-name	Represents the name of this machine within a distributed-architecture environment. This name is for use solely among the Connection Server installations on the machines.	Default: There is no default value.
MAC-address	In order to generate UUIDs for users, Connection Server needs a MAC address. Under most circumstances, Connection Server will automatically detect your MAC address. However, there may be instances where the Connection Server cannot detect it. In those situations, you can use this attribute to manually insert it.	Format: 6 pairs of hexadecimal digits, separated by colons: XX:XX:XX:XX:XX:XX Each **X** represents a hexadecimal digit **0-9** or **A-F** Example: 9f:6a:03:4f:0d:09
max-package-size	Specifies the maximum number of items that can be delivered in any given ICE package.	Default: 500
max-push-bps	Dictates the maximum rate at which the Connection Server will push ICE packages to subscribers.	Value is in bytes per second. A value of **-1** means no limit.
max-push-package-creation-threads	Specifies the maximum number of threads to use for generating packages to distribute via push.	*Default:* No limit is imposed.
max-push-retry	Specifies the maximum number of milliseconds that Connection Server waits before retrying a connection to Subscription Client when attempting a **push** mode delivery of content. The Connection Server, by default, makes several attempts at short time intervals prior to using the value in this configuration setting.	Default: 300000 (5 minutes)
max-push-threads	Maximum number of threads a **push** mode delivery opens.	Default: 6
min-push-threads	Minimum number of threads a **push** mode delivery keeps open.	Default: 3
push-connect-timeout	Specifies the amount of time in milliseconds allowed for push to connect to a client before timing out.	Default: 30000

Attribute	Purpose	Values and Default
package-flush-interval	The minimum interval at which filewatcher (and other content source manager) items can be added.	Default: 5 (seconds)
run-interactive	Specifies whether to open the Status Window at startup:	**true**: Open Status Window at startup **false**: Do not open Status Window at startup. Default: true
start-browser	Specifies whether to launch the web browser at startup (with the Administrator menu page displayed):	**true**: Launch web browser at startup **false**: Do not launch web browser at startup Default: true

This example of an **options** element sets two options:

```
<options browser-path="/usr/local/bin/netscape" content-browsing="false">
```

The timeFormat element

This attribute specifies parameters for the display and entry of time data. These settings are configurable on the Connection Server General Settings interface page.

Attribute	Purpose	Values and Default
use24hour	Specifies whether to use a 24-hour clock format or a 12-hour format	**true**: Use 24-hour format **false**: Use 12-hour format Default: false
time_format	Specifies the time to use, either **localTime** or **GMT**	Default: localTime
sign	Time can be specified as an offset of hours and minutes from GMT with a sign + or -.	+ adds hours/minutes to the GMT time - subtracts hours/minutes from the GMT time Default: blank
hours	Specifies the number of hours to add/subtract from the GMT time for the offset	Default: blank
minutes	Specifies the number of minutes to add/subtract from the GMT time for the offset	Default: blank

The log element

The table below shows the attributes for the log sub-element. The log element has no child elements.

Attribute	Purpose	Values and Default
default	Specifies the minimum priority of log messages written to the log. See "Logging levels" on page 28 for additional information on message priorities.	Default: info
overwrite	Selects whether new logs are appended to the existing log. Otherwise, the log file is overwritten each time Connection Server starts.	**true**: Overwrites the existing log file each time Connection Server starts **false**: Appends log messages to the existing log file. Default: false

Logging levels

You can set the level of events that are logged from various component facilities of Connection Server. The severity level values are:

- **debug:** for programmers
- **verbose:** multiple-line status messages
- **info:** informational messages
- **warning:** unusual conditions
- **error:** an operation has failed
- **critical**: serious error or crash

This example of a **log** child element sets the default logging level for all facilities and the logging level for two facilities:

```
<options>
  <log default="warning" ice="critical" replicator="debug"/>
</options>
```

This example sets the default logging level to **warning**, suppressing messages of lower severity and returning messages of higher severity. Messages from the ICE facility are suppressed unless they are of **critical** priority, while all events, including debugging information, are logged for the replicator facility.

In general, we recommend that you set logging levels for facilities by means of the Connection Server user interface, rather than by adding tags to the configuration file (as in the example above). If you do set these levels through the configuration file, your entries there override any logging levels that you set through the user interface. You should generally add *log default* tags to the configuration file only if you are trying to capture additional information about an error while starting the server.

Logging facilities

The following are the facilities for which event logging can be specified:

Facility	Description
analyzer	Component used by the webcrawler (**csm.web**) that analyzes web pages, JavaScript, etc., when crawling a site for download.
audit	Logs authorization events, and provides an audit trail of who changed what content .
auto-upgrade	Subscriber component that handles software updates delivered over the Internet.
csm.dirs	Content source monitor (csm) used to watch directories of files (also referred to as filewatcher).
csm.files	Content source monitor used to monitor sets of individual files (occasionally referred to as item level offers).
csm.web	Content source monitor used to monitor websites or ftp sites.
database	All of the server's basic interactions with the database are logged here (except for things like custom database content source monitors).
dataobject	Messages concerning the various entities (offers, subscriptions, packages) contained in the system and their interaction with the database.
date-time	All messages associated with date or time conversions are logged here.
delivery	Messages regarding the general aspects of content delivery, independent of delivery method.
delivery.ftp	Messages specific to delivering content via FTP.
delivery.ice	Messages specific to delivering content via ICE (note that both pull and push ICE subscriptions would log here).
delivery.mail	Messages specific to delivering content via e-mail.
ejb	Messages relating to the Connection Server's interaction with a J2EE application server.
event	Notification of the creation, modification, and deletion of resources (such as offers, subscribers, and subscriptions)—including the distribution of such events over the network.
filter	Messages generated by content filtering.
httpd	Connection and networking messages generated by Connection Server's built-in web server(s).
httpd.content	Messages generated by Connection Server's built-in Content Server (for serving files from content source monitors).
httpd.tomcat	Messages generated by Connection Server's administration server.
ice	The actual ICE messages exchanged between Connection Server and Subscription Client, as well as their processing.

Facility	Description
ice-cache	Messages concerning insertions and removals from the Connection Server's ICE cache (a subsystem used to queue up items which will be sent to the Subscription Client, optimizing delivery whenever possible).
KTL	Messages generated by the Kinecta Transformation Language. .
ldap	Messages generated by the LDAP synchronization subsystem while communicating with an LDAP server.
logging	Messages generated by the logging system itself, for logging to files such as databases and e-mail.
login	Logs user attempts to log on to the Connection Server.
replicator	Component used by csm.web to crawl websites; also used by the Subscription Client.
	Triggers and content sinks will log in this facility. Content sinks are modules for storing content in different repositories—the logical opposite of content source monitors.
scheduler	Records times of content updates, delivery, etc.
security	General messages from the security and authorization subsystem.
serializer	Messages generated by component that handles transaction boundaries for content source monitors.
soap	Describes SOAP (Simple Object Access Protocol) transaction activity.
sweeper	Cleans up after the Replicator.
syndicator	Messages produced by the general operation of the Connection Server itself (including most informational messages).
template	Messages produced by Connection Server's sending of e-mail messages defined by event templates.
ui	Messages from the administration JSP pages themselves (not the TomCat server).
xml	Our xml/html parser, used in various places but most notably for handling ice packages.

The proxy element

The table below shows the attributes for the proxy sub-element. The proxy element has no child elements.

Attribute	Purpose	Values and Default
host	Specifies the IP address or host name of the proxy server	Valid IP address or host name for the proxy server Default: null (-1)
password	Specifies the password for proxy authentication.	Valid password for secure proxy server.

Attribute	Purpose	Values and Default
port	Specifies the port number of the proxy server	Valid port number Default: null (-1)
proxyset	Specifies whether proxy server settings are set:	**true**: Pproxy server settings are set **false**: Proxy server settings are not set Default: false
ssl-port	Specifies the port number used by the proxy server for Secure Sockets Layer (SSL) requests.	Valid port number. Default: null (-1)
username	Specifies the user name for proxy authentication.	Valid user ID for secure proxy server.

Some HTTP proxy servers require a username and password. The Connection Server can use these proxy servers for ICE push delivery if you include the **username** and **password** tags in the configuration file's **proxy** element. The Connection Server supports HTTP proxy authentication for ICE push only. An example proxy element would look like this:

```
<options>
  <proxy host="proxy.stellent.com" port="3128" proxyset="true"
    username="user" password="password"/>
</options>
```

Note: The web crawler cannot do proxy authentication

The ssl element

The table below shows the attributes for the ssl (Secure Socket Layer) sub-element. The ssl element has no child elements.

Attribute	Purpose	Values and Default
enable	Specifies whether SSL security will be used by the Publishing Utility. Set this attribute to "true" to enable SSL.	**true**: The Publishing Utility will use SSL security to connect to the Administration user interface and the ICE server. **false**: The Publishing Utility can only use non-SSL ports. Default: false
required	Specifies whether only SSL security can be used and prohibits the use of other protocols.	**true**: Only SSL can be used. Non-SSL connection are not supported. **false**: Both SSL and non-SSL connections are supported. Default: false

Attribute	Purpose	Values and Default
admin	Specifies whether the Administration user interface is enabled over SSL. Set this attribute to "true" to enable this web-based interface.	**true**: The Administration user interface is enabled. **false**: The Administration user interface is not enabled. Default: false
keystorefilename	The SSL certificate file name. **yourfilename**: Provide your actual certificate file name.	
keystorepass	The SSL Keystore password. **yourpassword**: Provide your actual password.	

Secure Sockets Layer (SSL) certificates are not provided for you. You must create a SSL certificate and edit the configuration file to enable SSL. The configuration file for your instance will be either "cns.sqlserver.config" or "cns.oracle.config" (depending on your database). An example ssl element would look like this:

```
<options>
  <ssl enable="true" required="false" admin="true"
    keystorefilename="yourfilename" keystorepass="yourpassword"/>
</options>
```

The services element

This section covers the services element and following sub-elements:

- The admin-server attribute

- The content-server element

- The admin-server attribute

- The ice-server element

- The master-server element

In this example for configuring **services**, the ICE Server is auto-started and the other two servers are not. Port numbers are assigned:

```
<services>
  <ice-server port="8890" />
  <content-server start="false" port="8891"/>
  <admin-server start="true" port="8889" />
  <master-server receiver1="http://slave1:8890"/>
</services>
```

The admin-server attribute

This attribute specifies operational parameters for the Administration Server, which allows Connection Server to be administered through by the browser-based Administrator interface.

Attribute	Purpose	Values and Default
max-threads	Specifies the maximum number of request-handling threads this service can create; the number of threads specified is never exceeded.	Integer Default: 5
max-thread-queue	Specifies the maximum allowable number of backlogged requests.	Integer Default: 100
min-threads	Specifies the initial number of request-handling threads this service will create when started; the number of threads specified always start.	Integer Default: 1
port	Specifies the port on which that service should listen.	A valid port number Default: 8889
start	Specifies whether a given service should start automatically.	**true**: Service should start automatically **false**: Service should not automatically start Default: true

Note: Once **max-thread-queue** is exceeded, the service responds to new requests with the message: 500 Server Too Busy

The content-server element

Specifies operational parameters for the standard HTTP server that serves the content files.

Attribute	Purpose	Values and Default
master-url	Specifies the URL of the master host from which this Content Server instance receives event notification. If null or not specified, the service does not attempt to connect to a master.	URL:port Port default is 8880
max-threads	Specifies the maximum number of request-handling threads this service can create; the number of threads specified is never exceeded.	Integer Default: 30
max-thread-queue	Specifies the maximum allowable number of backlogged requests	Integer Default: 400
min-threads	Specifies the initial number of request-handling threads this service will create when started; the number of threads specified always start.	Integer Default: 2

Attribute	Purpose	Values and Default
port	Specifies the port on which the Content Server listens	A valid port number Default: 8891
remote-host	Specifies an external host that is used to serve content. URLs pointing to local content delivered by the Connection Server are changed to point to the host specified here, instead of the Content Server. The remote-host attribute can also be specified in the <options> element. This is a special use of the attribute that is related to backwards compatibility.	Valid host name Default: none
self-url	Specifies the URL and port the Content Server uses to receive event notifications from the master. Recommended for use when the IP address may not reliably point to the same machine, such as in some DHCP environments	URL:port If null or not specified, Connection Server determines a value, based on its current IP address.
ssl-port	Specifies the port on which the Content Server listens for communications protected by Secure Sockets Layer (SSL) security.	A valid port number Default: 8894
start	Specifies that the Content Server service should be started automatically	**true** or **false** Default: true (auto start)

The ice-server element

Specifies operational parameters for the ICE Server—the service that uses the ICE protocol to handle offers, subscriptions, and updates.

Attribute	Purpose	Values and Default
master-url	Specifies the URL of the master host from which this ICE Server receives event notification. If null or not specified, the service does not attempt to connect to a master.	URL:port Port default is 8880
max-threads	Specifies the maximum number of request-handling threads this service can create; the number of threads specified is never exceeded.	Integer Default: 7

Attribute	Purpose	Values and Default
max-thread-queue	Specifies the maximum allowable number of backlogged requests.	Integer Default: 30
min-threads	Specifies the initial number of request-handling threads this service will create when started; the number of threads specified always start.	integer Default: 7
port	Specifies the port on which the ICE Server service should listen.	A valid port number Default: 8890
self-url	Specifies the URL and port this host uses to receive event notifications from the master host. Recommended for use when the IP address may not reliably point to the same machine, such as in some DHCP environments.	A fully-qualified URL and port number. *Example:* **http://*your_url*:7575** If null or not specified, the master determines a value, based on the IP address.
ssl-port	Specifies the port on which the ICE Server listens for communications protected by Secure Sockets Layer (SSL) security.	A valid port number Default: 8893
start	Specifies that the ICE Server service should start automatically.	**true**: ICE Server starts automatically **false**: ICE Server does not automatically start Default: true

The master-server element

Specifies operational parameters for event distribution—the service that allows notifications to be sent between Connection Server components in multiple-host environments.

Attribute	Purpose	Values and Default
accept-unlisted	Specifies whether or not the master (event sending) host should allow unlisted slave (receiving) hosts to register for event notification. Unregistered slaves are not automatically reconnected when the master restarts and, therefore, are not automatically synchronized.	**true**: Unregistered slaves are accepted for connection to a master **false**: Only registered slaves are accepted for connection to a master Default: false
authentication-id	This attribute allows you to specify a login that the various servers in the distributed architecture will use to authenticate with each other.	Connection Server logins/usernames should contain only ASCII characters

Attribute	Purpose	Values and Default
authentication-pass	This attribute allows you to specify a password that the various servers in the distributed network will use to authenticate with each other.	Connection Server passwords should contain only ASCII characters.
master-url	Allows two master servers to communicate with each other, which might be necessary if changes are potentially happening on multiple machines.	A fully-qualified URL and port number.
port	Specifies the port on which the master host listens.	A valid port number Default: 8880
receivern	Specifies a list of URLs for event-receiving slave hosts. If accept-unlisted is false, each slave must be listed to receive events. This is the recommended configuration.	receiver*n*, where *n* is an integer value starting with **1**. Default: none
self-url	Allows two master servers to communicate with each other, which might be necessary if changes are potentially happening on multiple machines.	A fully-qualified URL and port number.
shutdown-slaves	Specifies whether or not the slaves should be notified when the master server shuts down or restarts. If true, the various slaves will also either be shutdown or restarted.	Default: true
start	Specifies that the master service should start automatically.	**true**: Master service starts automatically **false**: Master service does not automatically start Default: true

The database element

Most JDBC driver documentation details the Java code needed for a program to connect to a specific JDBC driver. In general, there are four common interactions. Each of these interactions can be specified in the configuration file; not all interactions are needed for each driver.

■ Selecting the JDBC driver to be used

■ Choosing how the driver is to be registered

■ Selecting the JDBC URL used to connect to the database

■ Setting additional parameters not specified in the JDBC URL

The database element and its child elements, driver, user, and property, enable Connection Server to connect to a database using a JDBC driver.

Attribute	Purpose	Values and Default
max-connections	The maximum number of database connections allowed.	whole number >0 Default: 9
max-retries	The number of times a database operation will be retried after one fails.	Default: 3
retry-delay	The amount of time in milliseconds between database retries.	amount of time in milliseconds between database retries

The description of these elements is shown in the following example. The database section of the configuration file appears in the configuration file provided with the Connection Server software.

```
<database type="dbtype" max-connections=9/>
  <driver jdbcURL="jdbc:oracle:kinecta_ice"
    driver="oracle.jdbc.driver.OracleDriver"/>
  <user username="user" password="pass"/>
  <property key="debug-level" value="2"/>
  <property key="keyname" value="5"/>
</database>
```

Attribute	Purpose	Values and Default
driver	The name of the JDBC driver.	The driver needs to be in your `classpath` or otherwise accessible from the environment in which Connection Server runs. .
jdbcURL	The JDBC URL used to connect to the database.	jdbcURL="jdbc:odbc:kinecta_ice" Properties can be set in the configuration file using the System.setProperties( ) method of the <property> element.
username	The user name to log into the database.	Default: username="user"
password	The user password to log into the database.	Default: password="pass"

The content-sources element

The content-source element allows the addition of a custom content source monitor. The factory element is a child of this element. For more information and an example, refer to the *Replication Customization Guide*.

The extensions element

This section covers the `extensions` element and following sub-element:

- The factory element

The **extensions** element is used to add custom extensions to the Connection Server. Its inclusion is wholly optional and most often will enable the addition of custom content source monitors and adapters. The SDK contains examples of such. The extensions element specifies a class that implements **com.kinecta.syndicator.extension**. This element has a child element: **factory** (any number, including none).

Attribute	Purpose	Values and Default
class	Java class name to be invoked for an extension.	valid Java class name for this extension
param	Information to be passed to the Java class when invoked.	any string you wish to pass to the extension

The factory element

The factory element implements com.kinecta.syndicator.extensionfactory and can be a child of both the extensions element and the content-sources element. It has no child elements.

Attribute	Purpose	Values and Default
class	Java class name to be invoked for a factory.	valid Java class name for this factory
param	Information to be passed to the Java class when invoked.	any string you wish to pass to the factory

The ldap element

The ldap element causes the Connection Server to launch in LDAP mode. This mode allows the Connection Server to import and delete user records by validation against an LDAP (Lightweight Directory Access Protocol) database.

Note: The Connection Server does not directly authenticate against the LDAP database

Attribute	Purpose	Values and Default
ldap-enabled	Tells the Connection Server whether to run in LDAP mode or not.	**true** or **false** (no default)
ldapURL	Specifies the LDAP server's IP address and port.	ldap://*your_IP_address*:389
username	Specifies an LDAP administrator user name, in order to establish a JNDI connection.	The user name with administrator permissions. For the iPlanet LDAP server, *default* is: "cn=Directory Manager"
userpassword	Specifies an LDAP administrator password, in order to establish a JNDI connection.	The password corresponding to the user name above.
rolebase	Specifies the LDAP element that forms the base of the search for matching roles.	Default: "cn=KinectaAdminGroup, ou=Groups, dc=*ldap_server_domain_name_element*,... dc=*ldap_server_domain_name_element*"
rolename	Specifies the name of the LDAP server attribute that contains the role name.	Example: "cn"
rolesearch	Specifies the LDAP search pattern for selecting roles in the LDAP realm.	Example: "cn" rolesearch="(uniquemember={0})"
digest	Specifies the digest algorithm used to store passwords.	Default: "CLEAR"

Attribute	Purpose	Values and Default
rolesubtree	Specifies whether to search subelements.	**true** if you want role searches to search the subtrees of elements selected by **rolebase**; **false** if you want to search only the top-level elements.
passwordname	Specifies the name of the LDAP server attribute that contains the password.	Form depends on the **digest** attribute's setting. Default: passwordname="userpassword"
userpattern	Specifies the search pattern for selecting users in the LDAP realm.	Use **{0}** as a shorthand for the distinguished name (dn) pattern corresponding to the users you want to retrieve. Example: userpattern="uid={0},ou=People, dc=*your_domain*,dc=com"

The j2ee element

This section covers the `j2ee` element and following sub-elements:

- The knet element

- The license-server host element

- The knet-server host element

- The soap-server host element

- The tracking-server host element

- The clickthru-server element

Attributes of the j2ee element

The **j2ee** element governs how the Connection Server behaves when run as a web application within the (J2EE-compliant) WebLogic application server. This element contains the following attributes:

Attribute	Purpose	Values and Default
initContextFactory	The name of the class that implements the naming context to facilitate the lookup and discovery of objects using JNDI services. This class name will vary depending on the JNDI services used.	
appServerURL	The complete URL (including the protocol and port) to access the application server. This URL is used to look up EJBs and other resources from the application server.	Default: t3://localhost:7001/
docRoot	The document root of the web application. The web application serves JSP and HTML pages to the client from this location.	Example: *installation_path*/webpages/ syndicator (where ***installation_path*** specifies the local directory in which the Connection Server is installed)
resourcePath	The root of the installed web application. The web application uses this path to look up other resources (such as configuration files).	Example: installation_path
ejbAccessUser	Together with **ejbAccessUserPasswd**, specifies the user credentials required to look up the EJBs in the application server.. If you change the System Administrator login/password in the web application, you should change these defaults to match.	Default: administrator
ejbAccessUserPasswd	Password corresponding to the value of **ejbAccessUser**.	Default: administrator
virtualDir	The name of the web application. Used for administrative purposes. Should match the name by which the web application is deployed in the application server.	Example: virtualDir=ConnectionServer

The knet element

The knet element directs the Connection Server to an appropriate Sellent Subscription Client Deployment Server. The knet element has five child elements: license-server host, knet-server host, soap-server host, tracking-server host, and clickthru-server.

The license-server host element

The **license-server host** child element specifies host information for the service that checks for licenses when the Connection Server and Subscription Client start up.

Attribute	Purpose	Values and Default
host	Specifies URL to the license-server host.	http://*your_url*
port	Specifies port used to access the license server.	Configurable, but usually shares the same port as the application server.

The knet-server host element

The knet-server host child element specifies host information for the Subscription Client Deployment Server itself.

Attribute	Purpose	Values and Default
host	Specifies the URL to the Subscription Client Deployment Server host.	http://*your_url*
port	Specifies port used to access the Subscription Client Deployment Server.	Configurable, but usually shares the same port as the application server.

The soap-server host element

The soap-server host child element specifies host information for the SOAP (simple Object Access Protocol) server. The SOAP protocol is used for communication between the Subscription Client and the Connection Server.

Attribute	Purpose	Values and Default
host	Specifies URL to the SOAP server's host.	http://*your_url*
port	Specifies port used to access the SOAP server.	Configurable, but usually shares the same port as the application server.

The tracking-server host element

The tracking-server host child element identifies the server that hosts the Content Matrix service.

Attribute	Purpose	Values and Default
host	Specifies URL to the Content Metrics server's host.	http://*your_url*
port	Specifies port used to access the Content Metrics server.	Configurable, but usually shares the same port as the application server.

The clickthru-server element

The clickthru-server child element identifies the server that hosts the Content Metrics click-through tracking service.

Attribute	Purpose	Values and Default
host	Specifies URL to the Content Metrics click-through server's host.	**http://***your_url*
port	Specifies port used to access the Content Metrics click-through server's host.	Configurable, but usually shares the same port as the application server.

CHAPTER 5

Using Content Server adapters

This chapter describes how to install the Content Server Adapters for integrations with Connection Server. There are two adapters, one residing on the server machine and the other on client machines:

■   The Outbound Adapter (also known as a content source) resides on a server machine. It allows the Connection Server to distribute content that is stored in Content Server. This relationship allows Content Server to function as the Connection Server's information store.

■   The Inbound Adapter (also known as a sink) resides on a client machine. It allows Subscription Client to populate Content Server with content that Subscription Client receives from any source. The sources will typically be other instances of Content Server, but other sources are possible.

By using these two adapters together, you can replicate content between Content Servers. However, the Inbound Adapter's architecture also allows you to use a single instance of Subscription Client to both receive content from, and store information in, a single instance of Content Server.

Note: See "Conventions used in this guide" on page 5 for information on path names and conventions used throughout this guide.

In this section:

■   Outbound Adapter

■   Inbound Adapter

Outbound Adapter

This section describes how to install Content Server Outbound Adapter for integrations with Connection Server. By installing this adapter, you can use the Connection Server to distribute content that originates in Content Server.

The section contains the following sections:

■   Installing the Outbound Adapter

■   Configuring Metadata Mapping

■   Using the Outbound Adapter

Notes

■   To successfully install the Outbound Adapter, you must install the Connection Server on the same machine as Content Server.

■   The prefix CNS_HOME/ refers to the Connection Server installation directory.

■   The prefix Content_Server_path/ refers to the Content Server installation directory.

Installing the Outbound Adapter

Both the Connection Server and Content Server must be configured and/or files edited within these server's directories to install the outbound adapter.

Connection Server Configuration

Perform the following steps to configure the Connection Server:

1. Make sure the $JAVA_HOME/lib/tools.jar file is in your CLASSPATH.

2. Open the Connection Server configuration file in a text-only editor. This file is located in the Connection Server installation directory.

   The Connection Server configuration file is named *cns.oracle.config* or *cns.sqlserver.config* depending on your database.

   You can copy prototype versions of these tags from the template file installed at CNS_HOME/extras/cns.contentserver.config.sample, and paste those into the Connection Server configuration file for further customization.

3. Edit the Connection Server configuration file:.

   - Add this tag to the options element to select which metadata fields from Content Server you want to send along with each content item:

     <options custom-item-fields=

     "ContentName,Title,Author,SecurityGroup,Type" />

     The comma-separated list in the above tag is an example. You can modify this list to determine which fields to send. You should omit the leading x from any custom field names from the IDC schema. See "Configuring Metadata Mapping" on page 48 for additional information.

   - Add this tag to load the outbound adapter:

     <content-sources>

     <factory class=

     "com.kinecta.adapter.contentserver.syndicator.

     StellentAdapter" param="Content_Server_path/cns_queue"/>

     </content-sources>

   - Specify a queue directory. Content Server will use this directory to queue updates that occur when the Connection Server is not running, or updates for which no offer has yet been created in the Connection Server. When the Connection Server is relaunched or provided with an appropriate offer, it will process this queue

     The default queue directory is Content_Server_path/cns_queue. Since the Connection Server does not know where Content Server is installed, you must specify the full path to this directory even if you use this default value

4. Start the Connection Server.

5. To begin deploying the adapter, point a browser to the SOAP admin tool:

   For example:

   http://testserver:8889/soap/

   This URL assumes that you are using the default 8889 administrative port for a standalone installation.

6. For the admin client, select **Run** and then **Deploy**.

7.   To complete the deployment, use the SOAP admin tool to set these values:

Field	Value
ID	tag:stellent,2002-06:scs-cns/soap-enc-ns
Scope	request
Methods	updateContent
Java Provider Class	com.kinecta.adapter.contentserver.syndicator.SoapReceiver
Static?	no

Note: Be sure to select a Java Provider Class, not a Java Provider Type.

Content Server Configuration

Perform the following steps within Content Server installation directory.

1.   Edit the Content Server *config.cfg* file and add the CNSSoapUrl value appropriate to your Connection Server installation:

For example:

CNSSoapUrl=http://localhost:8889/soap/servlet/rpcrouter

The above example assumes that you are using the default administration port 8889. If you have configured a different administration port, change the 8889 string within this URL accordingly.

2.   Edit the Content Server *config.cfg* file and add the following values:

CNSSoapUrn=tag:stellent,2002-06:scs-cns/soap-enc-ns

CNSSoapUser=administrator

CNSSoapPass=administrator

CNSSendPrimary=[true|false]

CNSSendWeblayout=[true|false]

CNSSendAlternates=[true|false]

CNSSendRenditions=[true|false]

Enter that user's identifiers in the CNSSoapUser and CNSSoapPass lines that you defined earlier. The [true|false] options determine which files Content Server sends to Connection Server.

3.   For security reasons, you might want to substitute a user account and password other than administrator as the CNSSoapUser and CNSSoapPass.

Create a new user in Connection Server with these permissions:

ContentPackage Write System

ContentPackage Delete System

ContentPackage Create System

4.   From the CNS_HOME/lib directory, copy the activation.jar, soap.jar, and mail.jar files into the Content_Server_path/shared/classes directory.

5.   Edit the Content_Server_path/bin/intradoc.cfg file and add the jar files (activation.jar, soap.jar, and mail.jar) to that file's CLASSPATH statement.

When editing the intradoc.cfg file CLASSPATH statement: Regardless of what platform you are on, you should follow the path conventions already in use in the file. For example, even on Windows platforms, refer to soap.jar as $SHAREDDIR/classes/soap.jar.

6. Upload the CNS.zip component (distributed with Connection Server and located at: CNS_HOME/extras/CNS.zip) to Content Server and activate it.

To perform the upload using the Admin Server Component Manager page

1. Log into Content Server as the system administrator and click **Administration**.

2. Under Administration Pages, click **Admin Server**.

3. Click the desired Content Server instance (for example, Master_on_testserver).

4. Under Options, click **Component Manager**.

5. The Component Manager page appears.

6. In the Upload New Component section, click **Browse**.

7. Navigate to the CNS.zip component and click **Open**.

8. Click **Upload**.

9. From the Disabled Components list, select CNS and click **Enable**.

10. Under Options, click **Stop/Start Content Server**. The Content Server Current Status page appears.

11. Stop and restart Content Server.

Configuring Metadata Mapping

In some cases, you will need to configure custom metadata mapping. See "Metadata Mapping Requirements" on page 49 for additional information.

This configuration can occur in up to three places:

■ In the Content Server Component Wizard. See "About Metadata Mapping" on page 48 for additional information.

■ In the Connection Server configuration file, by editing the options element custom-item-fields tag. See "Connection Server Configuration" on page 46 for additional information.

■ In the Subscription Client configuration file. See "Installing the Inbound Adapter" on page 51 for additional information.

About Metadata Mapping

The Content Server Outbound Adapter uses metadata mapping to map Content Server field names (for example, dDocTitle and dDocType) into normalized names (for example, Title and Type). The Connection Server tracks these normalized names and sends the corresponding fields in the ICE package.

This mapping is configured in the Content Server Component Wizard, in the Custom HTML Included section. The upload_xml_insert_structure resource maps items that are being added or updated, and the upload_xml_delete_structure resource maps deletions.

You can edit these resources by matching the format of their default contents. The name of each XML tag is a normalized field name for Connection Server; the value within the tag is the name of the corresponding Content Server field.

For example, this entry maps the Content Server field name dDocName to ContentName:

```
<ContentName>&lt;$dDocName$&gt;</ContentName>
```

By default, the upload_xml_insert_structure resource contains the following entries:

```
<OpType>Update</OpType>
<ContentName>&lt;$dDocName$&gt;</ContentName>
<Title>&lt;$dDocTitle$&gt;</Title>
<Type>&lt;$dDocType$&gt;</Type>
<Author>&lt;$dDocAuthor$&gt;</Author>
<SecurityGroup>&lt;$dSecurityGroup$&gt;</SecurityGroup>
<Accounts>&lt;$dDocAccount$&gt;</Accounts>
<RevisionID>&lt;$dRevisionID$&gt;</RevisionID>
<CustomFields></CustomFields>
<ContentServer>&lt;$IDC_Name$&gt;</ContentServer>
```

In the above example, the CustomFields entry is a special tag that is handled differently than all the other tags in this structure. While the other tags each represent a single metadata field and its value, the CustomFields tag represents all of the custom metadata fields that you have defined on Content Server.

The CustomFields tag will be replaced with the names and values of these custom metadata fields. This replacement occurs when the Content Server component uses the upload_xml_insert_structure resource to build up the list of metadata items to send to the Connection Server.

The names of the custom metadata fields are each prefixed with an x (as in: xCustom_Name), but when Connection Server sends them, it omits the x (as in: Custom_Name).

For example, if the upload_xml_insert_structure on Content Server contained the foloowing entries and you had defined custom fields xColor and xShape, then the Outbound Adapter would tell the connections server that dDocName is called ContentName, xColor is called Color, and xShape is called Shape:

```
<ContentName>&lt;$dDocName$&gt;</ContentName>
<CustomFields></CustomFields>
```

Metadata Mapping Requirements

Certain metadata mapping conditions must be met for the Outbound Adapter to function properly. Following the default installation instructions in this guide will meet these requirements. However, if you customize your content distribution, you may need to configure the metadata mapping to preserve these conditions:

- Certain fields must be present for batch loads to work on Subscription Client's Inbound Adapter. For a list of these fields, refer to the Content Server System Administration Guide section titled "Understanding Batch Load Files."

- If you want to add or delete fields, you must configure the mappings accordingly.

- If you are using both the Outbound Adapter and Inbound Adapter, their respective mappings must match.

- In the Inbound Adapter, you configure metadata mapping in the Subscription Client configuration file. The mapping is governed by the sink element's metadata-map tag.

- The Inbound Adapter's default mapping values do the opposite of the server-side mappings: They map the normalized values (which Connection Server sends in ICE packages) back into Content Server–specific names.

- The metadata-map tag's format is similar to that of entries within Content Server Component Wizard resources.

For example, this tag maps the ContentName property into dDocName:

<metadata-map ContentName="dDocName">

Using the Outbound Adapter

This section describes how to create an offer in the Connection Server once you have installed the Subscription Client. The procedure is similar to creating any other offer in the Connection Server, except that Content Server functions as the offer's content source.

Use the following steps to create an offer using the Content Server adapter:

1. Edit the Content Server *config.cfg* configuration file and add the following value:

    CNSArchives=Archiver_job_name

    Replace the Archiver_job_name placeholder with an arbitrarily chosen name. Alternatively, you can list multiple Archiver jobs, separated by commas.

    For example:

    CNSArchives=apple,banana,cantaloupe

2. Use the Connection Server interface to create an offer—navigate to the Connection Server Create Offer page.

3. As the Content Source Type, choose **Content Server**.

4. Click **Next** to move to the second Create Offer page.

5. As the Archiver Job Name, enter the name that you selected earlier.

6. Save the offer.

7. Create the corresponding Archiver job. You can do this from either the Connection Server or Content Server, as follows.

To create the Archiver job from the Connection Server

1. From the second Create Offer page, click **Launch Archiver**.

    If this is the first offer you have created since installing the adapter, the Connection Server will automatically display the *Choose Content Server* page.

2. On the *Choose Content Server* page fill in the **Content Server Instance** field.

    The Content Server Instance is the name of the Content Server to which you are connecting. For example: IDCNAME_ON_SERVER. This name was chosen upon Content Server installation.

3. Fill in the **Content Server Webroot** field.

    The Content Server Webroot is the name of the webroot directory (without slashes). For example, if you access Content Server as *http://testserver.test.com/idc/portal.htm* the Content Server webroot would be "idc". The webroot name was also chosen upon Content Server installation.

4. Click **Next** to continue.

    If this is not the first offer you have created since installing the outbound adapter, you can still display the Choose Content Server page if you want to edit these settings. To display this page from the second Create Offer page, click the **Edit Settings for Content Server** link.

To create the Archiver job from Content Server

1. Start the Archiver and create an Archiver job that covers the documents you want to distribute.

2. Name the Archiver job identically to the name that you selected earlier.

3. Make sure that Export Automated is set to Yes, and that the Content Server with the CNS component installed is a registered exporter (on the Archiver Replication tab).

   If you create an Archiver job in Content Server before creating a Connection Server offer that references that Archiver job name, Content Server will queue up messages in its queue subdirectory until you create an offer to distribute them.

4. With the offer that you have now created, you can create subscriptions to start delivering content.

Inbound Adapter

This section describes how to install the Content Server Inbound Adapter. Installing this adapter allows the Subscription Client to receive content that originates in the Content Server and is distributed via the Connection Server.

The section contains the following sections:

- Installing the Inbound Adapter

- Using the Inbound Adapter

Notes

- To successfully install the Inbound Adapter, you must install Subscription Client, on the same machine as the Content Server.

- The prefix Subscription_Client_path/ refers to the Subscription Client installation directory.

Installing the Inbound Adapter

The Subscription Client must be configured and/or files edited within this server's installation directory to install the inbound adapter.

Subscription Client Configuration

Perform the following steps within the Subscription Client installation directory.

1. Edit the Subscription Client configuration file to add the tags listed below:

Note: You can copy prototype versions of these tags from the template file installed at Subscription_Client_path/extras/siclone.config.sample, and paste those into the siclone.config file for further customization.

```
<defaults>
<sink factory="com.kinecta.adapter.contentserver.subscriber.
StellentSinkFactory">
<metadata-map ContentName="dDocName"
Title="dDocTitle"
Author="dDocAuthor"
SecurityGroup="dSecurityGroup"
```

```
Type="dDocType" />
</sink>
<sink factory="com.kinecta.util.FileSystemSink"/>
<trigger when="after" type="cmd">
batchload.sh %%asubscription-id%
</trigger>
<ice-delivery-rule/>
</defaults>
```

- For Windows, within the trigger tag, replace the batchload.sh reference with: batchload.bat.

- Depending on your installation, you might need to further customize the sample metadata-map attribute shown above.

The code above assumes that you want all of this Subscription Client's subscriptions to go into the Content Server. It therefore places the sink, metadata-map, sink factory, trigger, and ice-delivery-rule tags within the configuration file's defaults element. If you prefer to insert subscription packages into the Content Server selectively, you would instead place these tags inside the corresponding subscriptions' job elements. See "Sending Specific Subscriptions to Content Server" on page 52 for additional information.

2. Edit the batchload script to point to your installed copy of the Content Server and of the batchload.txt file. In the per-platform examples below, replace the paths in the IDC_BASE and BatchLoader entries with your actual installed paths.

Solaris—sample batchload.sh script:

```
#!/bin/sh
IDC_BASE=/space/stellent/server
PATH=$PATH:$IDC_BASE/bin
PWD='pwd'
BatchLoader -q -n $PWD/batchload.txt
mv batchload.txt batchload.old
```

Windows—sample batchload.bat script:

```
SET IDC_BASE=c:\stellent
SET PATH=%PATH%;%IDC_BASE%\bin
cd %IDC_BASE%\bin
BatchLoader.exe /q /console /n
"c:\Program Files\Stellent\Stellent Subscription Client\
batchload.txt"
cd "c:\Program Files\Stellent\Stellent Subscription Client"
rename batchload.txt batchload.old
```

Note: In this Windows example, the BatchLoader.exe entry is actually a single line.

Using the Inbound Adapter

This section provides information on sending specific subscriptions to the Content Server and metadata mapping.

Sending Specific Subscriptions to Content Server

The section "Subscription Client Configuration" on page 51 assumes that you want all of your Subscription Client subscriptions to go into the Content Server. The configuration file example therefore places the sink, metadata-map, sink factory, trigger, and ice-delivery-rule tags within

the configuration files default element. This element affects all subscriptions on the Subscription Client.

If you prefer to insert subscription packages into the Content Server selectively, you would instead place these tags inside the corresponding subscriptions' job elements within the configuration file.

Metadata Mapping and the Subscription Client

Within the Subscription Client configuration file, the values in the metadata-map attribute must match the values that are being sent in the ICE package. The sample values used in this guide are default values that would be sent from a Connection Server that is serving files from a Content Server. See "Metadata Mapping Requirements" on page 49 for additional information on matching the metadata mapping on the Inbound Adapter and Outbound Adapter.

If you are populating the Content Server with content that your Subscription Client has received from a source other than the Content Server, you must insert additional tags to add metadata. See "Metadata Mapping Requirements" on page 49 for additional information on the appropriate format for these tags.

THIRD PARTY LICENSES

OVERVIEW

This appendix includes a description of the Third Party Licenses for all the third party products included with this product.

- ❖ Apache Software License (page A-55)
- ❖ W3C® Software Notice and License (page A-56)
- ❖ Zlib License (page A-57)
- ❖ General BSD License (page A-58)
- ❖ General MIT License (page A-59)
- ❖ Unicode License (page A-59)
- ❖ Miscellaneous Attributions (page A-60)

APACHE SOFTWARE LICENSE

```
* Copyright 1999-2004 The Apache Software Foundation.

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*      http://www.apache.org/licenses/LICENSE-2.0

*
```

```
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

 * See the License for the specific language governing permissions and

 * limitations under the License.
```

W3C® SOFTWARE NOTICE AND LICENSE

```
* Copyright © 1994-2000 World Wide Web Consortium,

* (Massachusetts Institute of Technology, Institut National de

* Recherche en Informatique et en Automatique, Keio University).

* All Rights Reserved.  http://www.w3.org/Consortium/Legal/

*

* This W3C work (including software, documents, or other related items) is

* being provided by the copyright holders under the following license. By

* obtaining, using and/or copying this work, you (the licensee) agree that

* you have read, understood, and will comply with the following terms and

* conditions:

*

* Permission to use, copy, modify, and distribute this software and its

* documentation, with or without modification, for any purpose and without

* fee or royalty is hereby granted, provided that you include the following

* on ALL copies of the software and documentation or portions thereof,

* including modifications, that you make:

*

*   1. The full text of this NOTICE in a location viewable to users of the

*      redistributed or derivative work.

*

*   2. Any pre-existing intellectual property disclaimers, notices, or terms

*      and conditions. If none exist, a short notice of the following form

*      (hypertext is preferred, text is permitted) should be used within the

*      body of any redistributed or derivative code: "Copyright ©

*      [$date-of-software] World Wide Web Consortium, (Massachusetts
```

```
*      Institute of Technology, Institut National de Recherche en

*      Informatique et en Automatique, Keio University). All Rights

*      Reserved. http://www.w3.org/Consortium/Legal/"

*

*   3. Notice of any changes or modifications to the W3C files, including the

*      date changes were made. (We recommend you provide URIs to the location

*      from which the code is derived.)

*

* THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS

* MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT

* NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR

* PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE

* ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

*

* COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR

* CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR

* DOCUMENTATION.

*

* The name and trademarks of copyright holders may NOT be used in advertising

* or publicity pertaining to the software without specific, written prior

* permission. Title to copyright in this software and any associated

* documentation will at all times remain with copyright holders.

*
```

ZLIB LICENSE

```
* zlib.h -- interface of the 'zlib' general purpose compression library
  version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler
This software is provided 'as-is', without any express or implied
  warranty.  In no event will the authors be held liable for any damages
  arising from the use of this software.
```

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org

Mark Adler madler@alumni.caltech.edu

GENERAL BSD LICENSE

Copyright (c) 1998, Regents of the University of California

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

"Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

"Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

"Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GENERAL MIT LICENSE

Copyright (c) 1998, Regents of the Massachusetts Institute of Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNICODE LICENSE

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/ . Unicode Software includes any source code published in the Unicode Standard or under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/.

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2006 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

_____Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions. All other trademarks and registered trademarks mentioned herein are the property of their respective owners

MISCELLANEOUS ATTRIBUTIONS

Adobe, Acrobat, and the Acrobat Logo are registered trademarks of Adobe Systems Incorporated.

FAST Instream is a trademark of Fast Search and Transfer ASA.

HP-UX is a registered trademark of Hewlett-Packard Company.

IBM, Informix, and DB2 are registered trademarks of IBM Corporation.

Jaws PDF Library is a registered trademark of Global Graphics Software Ltd.

Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.

Linux is a registered trademark of Linus Torvalds.

Mac is a registered trademark, and Safari is a trademark of Apple Computer, Inc.

Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation.

MrSID is property of LizardTech, Inc. It is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Oracle is a registered trademark of Oracle Corporation.

Portions Copyright © 1994-1997 LEAD Technologies, Inc. All rights reserved.

Portions Copyright © 1990-1998 Handmade Software, Inc. All rights reserved.

Portions Copyright © 1988, 1997 Aladdin Enterprises. All rights reserved.

Portions Copyright © 1997 Soft Horizons. All rights reserved.

Portions Copyright © 1995-1999 LizardTech, Inc. All rights reserved.

Red Hat is a registered trademark of Red Hat, Inc.

Sun is a registered trademark, and Sun ONE, Solaris, iPlanet and Java are trademarks of Sun Microsystems, Inc.

Sybase is a registered trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group.

Verity is a registered trademark of Autonomy Corporation plc

Index