Universal Content Management
Version 10gR3

Security Providers
Component
Administration Guide
Copyright © 2008 Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software- -Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.
## Table of Contents

### Chapter 1: Introduction
- Overview ........................................................................ 1-1
- Product Overview ....................................................... 1-1
- Requirements ................................................................. 1-2
- Component Contents ..................................................... 1-2
- Audience ...................................................................... 1-3
- About This Guide ......................................................... 1-3

### Chapter 2: Installation
- Overview ........................................................................ 2-1
- Installing The Security Providers Component .................. 2-1

### Chapter 3: Using Security Providers
- Overview ........................................................................ 3-1
- About Security Providers............................................... 3-1
- Planning: ........................................................................ 3-3
  - Keepalive Connections ............................................... 3-3
  - SSL Connections ....................................................... 3-4
  - Additional Configuration ............................................. 3-5
- Keystores and Truststore ................................................ 3-5
  - When to Use Keystores and a Truststore ..................... 3-6
  - Specifying Keystore and Truststore Information ........... 3-6
  - Generating a Keystore ................................................. 3-6
  - Creating a Truststore ................................................ 3-8
- Managing SSL and Keepalive Providers ......................... 3-9
  - Adding an Incoming Security Provider ....................... 3-9
  - Adding an Outgoing Security Provider ....................... 3-10
Table of Contents

Security Provider Interface Pages ....................................3-11
  Providers Page ......................................................3-11
  Add keepaliveincoming Provider Page .................................3-13
  Add keepaliveoutgoing Provider Page .................................3-14
  Add sslincoming Provider Page ....................................3-17
  Add ssloutgoing Provider Page ....................................3-19

Index
INTRODUCTION

OVERVIEW

This chapter covers the following topics:

- Product Overview (page 1-1)
- Requirements (page 1-2)
- Component Contents (page 1-2)
- Audience (page 1-3)
- About This Guide (page 1-3)

PRODUCT OVERVIEW

The Security Providers component provides additional security for the Content Server by extending the functionality of basic socket providers with two new types of providers:

- Secure socket layer (SSL) providers
- Keepalive providers
Benefits of using the Security Providers component include the following:

- SSL enhances security for web communication by providing communication encryption and authentication.
- Security providers enable use of certificates for socket or server authentication.
- Keepalive and connection pooling logic help avoid SSL expense overhead by reducing the amount of SSL socket creation and teardown.

**Requirements**

The following items are important to consider when deciding to use the Security Providers component:

- This component runs on Content Server under the supported Windows and UNIX operating systems.
- This documentation discusses the component as it runs on Content Server version 10gR3.

**Component Contents**

The Security Providers component file, SecurityProvider.zip, is available for download and is included with the samples and extras. It contains the following files:

<table>
<thead>
<tr>
<th>Description</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component files</td>
<td>*.hda</td>
</tr>
<tr>
<td></td>
<td>*.class</td>
</tr>
<tr>
<td></td>
<td>*.htm</td>
</tr>
<tr>
<td></td>
<td>*.zip</td>
</tr>
</tbody>
</table>
AUDIENCE

This guide is intended for people who design, install, or administer the network connections and providers for server and client communication between the Content Server and web browsers. These architects and administrators must be familiar with networking and security concepts, and with Content Server services and processes.

ABOUT THIS GUIDE

This guide provides instructions to install and configure the Security Providers component on the Content Server. The information contained in this document is subject to change as the product technology evolves and as hardware, operating systems, and third-party software are created and modified.

Conventions

The following conventions are used throughout this document:

- The notation `<install_dir>/<instance>` is used to refer to the location on your system where a specific instance of Content Server is installed.
- Forward slashes (`/`) are used to separate parts of an Internet address. For example, http://www.microsoft.com/windows2000/. A forward slash might or might not appear at the end of an Internet address.
- Backward slashes (`\`) are used to separate the levels in a path to a server, directory, or file. For example, C:\stellent\. This is true when referring to files on a Windows file system or on a UNIX system. A backward slash will always appear after the end of a server, directory, or file path.
- Paths to access operating system dialogs or windows use the following formatting structure:
  
  **Start—Settings—Control Panel**

- Required user input is distinguished using the following font formatting:

  `xyz_name`
Introduction
Chapter 2

INSTALLATION

OVERVIEW

This chapter covers the following topics:

- Installing The Security Providers Component (page 2-1)
- (page 2-3)

INSTALLING THE SECURITY PROVIDERS COMPONENT

To install and enable the Security Providers component on Stellent Content Server, use either the Component Wizard or the Component Manager.

Component Wizard Installation

Use this procedure to install the component using the Component Wizard.

1. Obtain the Security Providers component file, SecurityProviders.zip, from the component update bundle for release 10g. Place the file in a temporary location.


   The Component Wizard main screen and the Component List screen are displayed.

3. On the Component List screen, click Install.

   The Install screen is displayed.
4. Click **Select**. Navigate to the location where you downloaded the SecurityProviders.zip file and select it.

5. Click **Open**.

   The zip file contents are added to the Install screen list.

6. Click **OK**.

   The Component Wizard prompts whether you want to enable the Security Providers component.

7. Click **Yes**.

   The Security Providers component is listed as enabled on the Component List screen.

8. Restart the content server.

---

### Component Manager Installation

Use this procedure to install the component using the Component Manager.

1. Obtain the Security Providers component file, SecurityProviders.zip, from the component update bundle for release 10g. Place the file in a temporary location.

2. Open a new browser window and log in to your Stellent Content Server as the system administrator.

3. Open the Administration tray.

4. Click **Admin Applets** to open the Administration page.

5. Click **Admin Server**.

6. Click the applicable content server instance.

7. In the sidebar click **Component Manager**.

   The Component Manager screen is displayed.

8. Select the Browse button next to the Install New Component box. Navigate to the SecurityProviders.zip file and select it.

9. Click **Install**.

   A page is displayed listing the component items that will be installed.

10. Click **Continue** to continue with the installation.

    A Content Server message is displayed that indicates the installation is successful.

11. On the message page click to return to the Component Manager.
12. Click the component name in the Disabled Components box.

13. Click **Enable** to enable the component.

   The component is listed in the Enabled Components box.

14. In the sidebar click **Start/Stop Content Server**.

15. Restart the Content Server.
Installation
ABOUT SECURITY PROVIDERS

The Security Providers component enables Secure Socket Layer (SSL) encryption and authentication for standard Stellent Content Server incoming and outgoing socket providers. The component also enables keepalive functionality for socket providers, which minimizes the creation and teardown of SSL sockets. Appropriate use of security providers, along with keys and certificates, can improve the security for network and Internet communication with the Content Server.

To use the Security Providers component it is necessary to be familiar with socket providers, security and authentication, SSL, keepalive, and other aspects of security for network communication. The following sources of information can be useful in working with the Security Providers component:

- Stellent Content Server Managing System Settings and Processes
- Stellent Content Server Managing Security and User Access
Using Security Providers

  This online document is available from Sun at www.sun.com. It contains an extensive Related Documents section that includes web links to reference books, security standards, government security policies and regulations, and a list of books on cryptography and SSL.
- keytool Key and Certification Management Tool
  This online document is available from Sun at www.sun.com.
- RSA's Public Key Cryptography Standards.
  This online document is available from RSA at www.rsasecurity.com.
- RSA’s Cryptography FAQ.
  This online document is available from RSA at www.rsasecurity.com.
- SSL Certificate FAQ.
  This online document is available from The Linux Documentation Project at www.tldp.org.

The following list includes definitions for some of the terms used in this guide. For detailed information refer to the previous list of information sources or to security and authentication standards sources.

- Certificate—A digital signature that verifies the identity and public key for an entity (a person or company). A certificate can be issued by a Certification Authority or by an individual entity.
- Certificate Authority (CA)—An entity that issues certificates for other entities, and is recognized as a well-known and trusted source for certificates, such as VeriSign and Thawte.
- Keystore—A file or database of information for keys, used for authentication processing.
- Private key—Information packaged as a key that is known only to the entity that issues it. Private keys are used in generating signatures.
- Public key—Information packaged as a key that is publicly associated with an entity. Public keys are used in verifying signatures.
SSL—Secure Socket Layer, a protocol for secure network communication using a combination of public and secret key technology.

Truststore—A file or database of keys that the trust manager has determined can be trusted.

Planning

It is recommended that you determine how you want to use security providers before implementing SSL socket providers or keepalive socket providers. Examine the keepalive and SSL connection types and determine whether additional configuration is required to use the security providers you select, such as a need to create keystores or a truststore. Refer to the additional sources of information listed in About Security Providers (page 3-1).

The following sections provide more information about the SSL and keepalive provider types, including the Java classes used to control the behavior of the provider types, and additional configuration that may be necessary.

- Keepalive Connections (page 3-3)
- SSL Connections (page 3-4)

Keepalive Connections

The keepalive feature enables persistent connections and the pooling of socket connections for service requests. The setup for keepalive connections is most useful in situations where connection setup and teardown can take a considerable amount of time, and you want to minimize the time spent on that activity. The Security Providers component provides two keepalive socket providers: incoming and outgoing.

The following Java classes are used to set up the keepalive incoming socket provider:

<table>
<thead>
<tr>
<th>Provider Class</th>
<th>idc.provider.ExtendedSocketIncomingProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Class</td>
<td>idc.provider.KeepaliveSocketIncomingConnection</td>
</tr>
<tr>
<td>Server Thread Class</td>
<td>idc.server.KeepaliveIdcServerThread</td>
</tr>
</tbody>
</table>
The following Java classes are used to set up the keepalive outgoing socket provider:

<table>
<thead>
<tr>
<th>Provider Class</th>
<th>idc.provider.KeepaliveSocketOutgoingProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Class</td>
<td>idc.provider.KeepaliveSocketOutgoingConnection</td>
</tr>
<tr>
<td>Request Class</td>
<td>idc.provider.KeepaliveServerRequest</td>
</tr>
</tbody>
</table>

**SSL Connections**

The SSL provider setup enables the use of SSL connections in a keepalive environment. This setup is recommended over a simple SSL provider setup because it helps minimize the cost of SSL socket setup and teardown. The Security Providers component provides two SSL socket providers with keepalive: incoming and outgoing.

The following Java classes are used to set up the SSL keepalive incoming socket provider:

<table>
<thead>
<tr>
<th>Provider Class</th>
<th>idc.provider.ssl.SSLSocketIncomingProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Class</td>
<td>idc.provider.KeepaliveSocketIncomingConnection</td>
</tr>
<tr>
<td>Server Thread Class</td>
<td>idc.server.KeepaliveIdleServerThread</td>
</tr>
</tbody>
</table>

The following Java classes are used to set up the keepalive SSL outgoing socket provider:

<table>
<thead>
<tr>
<th>Provider Class</th>
<th>idc.provider.KeepaliveSocketOutgoingProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Class</td>
<td>idc.provider.ssl.SSLSocketOutgoingConnection</td>
</tr>
<tr>
<td>Request Class</td>
<td>idc.provider.KeepaliveServerRequest</td>
</tr>
</tbody>
</table>
Additional Configuration

Depending on which type of security provider you choose, there can be additional required configuration.

- **Keepalive and SSL outgoing providers**—The Add Provider page includes a Num Connections field, which specifies the number of connections to pool.

- **SSL incoming providers**—The Add Provider page includes two additional options:
  - Request Client Auth option—If clients are able, they should authenticate themselves when they make a connection.
  - Require Client Auth option—Clients must authenticate themselves in order to make a connection.

SSL providers may also require setup of a keystore or keystores, and a truststore, for both the client and server, depending on the value of the Request Client Auth option, the value of the Require Client Auth option, and what type of Certification Authority signed the certificates handled by these options. For information on keystores and truststore refer to Keystores and Truststore (page 3-5).

Keystores and Truststore

SSL providers may require use of keystores and may require a truststore. Keystores are files that hold public and secret key information for use in SSL. A truststore contains certificates that have been determined to be trusted. If a certificate used on the server and client is signed by a well-known Certification Authority (CA) such as VeriSign or Thawte, then a truststore isn’t necessary, because the default JVM truststore contains the certificates of these CAs. Truststores are needed when certificates used by the SSL providers are self-signed or signed by a private CA. If SSL providers require keystores, and a truststore, then they must be created and managed.

The following sections provide overview information about keystores and truststore.

- When to Use Keystores and a Truststore (page 3-6)
- Specifying Keystore and Truststore Information (page 3-6)
- Generating a Keystore (page 3-6)
- Creating a Truststore (page 3-8)

For detailed information on keystores and truststores refer to the sources of information listed in About Security Providers (page 3-1).
When to Use Keystores and a Truststore

The following examples present different situations and uses for keystores and a truststore.

- The server requires a keystore containing a signed SSL certificate in order to create SSL sockets.
- The server requests or requires client authentication, which may require a truststore. If the client’s certificate is not signed by a well-known CA, then the server will need a truststore containing that CA’s certificate.
- The server requests or requires client authentication, which may require that the client have a keystore in which it stores the certificate the client presents for authentication.
- The server uses a certificate that hasn’t been signed by a well-known CA, therefore the client will require a truststore that contains the server’s certificate.

Specifying Keystore and Truststore Information

In order to use keystore and truststore information, the SSL incoming and outgoing providers require that a file named sslconfig.hda be set up in the providers directory (next to the provider.hda file). The sslconfig.hda file contains configuration information you specify for your keystore and truststore. It has a format similar to the following example. For security reasons, there is no web interface to assist in editing this file; all edits must be done manually using a text editor.

```
@Properties LocalData
TruststoreFile=/servers/idc/data/providers/ssloutgoing1/truststore
KeystoreFile=/servers/idc/data/providers/ssloutgoing1/keystore
@end
```

<table>
<thead>
<tr>
<th>Configuration Name</th>
<th>Value Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TruststoreFile</td>
<td>The full path to the truststore file.</td>
</tr>
<tr>
<td>KeystoreFile</td>
<td>The full path to the keystore file.</td>
</tr>
</tbody>
</table>

Generating a Keystore

This section describes the basic process for generating a keystore. You will need to determine the specific requirements and names for keys and keystores you want to create for your SSL providers. You can store keystore files wherever you want, because the sslconfig.hda file contains full paths for its KeystoreFile config settings. However, it is recommended that keystore files are stored in the
For detailed information on how to use the keytool utility to generate a keystore, see the document titled *keytool Key and Certification Management Tool*, available online from Sun at [www.sun.com](http://www.sun.com).

**Note:** The Java keytool utility has a feature that prevents direct interaction with private keys. This means a certificate that is generated using keytool is “stuck” in the keystore; there is no way to retrieve the private key portion of the certificate. Inversely, there is no way for keytool to import a pre-existing certificate into a Java keystore.

To use keytool you must have the utility in your path when you enter the command.

1. Create a key in a keystore. The following command-line example shows how to create a key entry with the name *alias* in a keystore with the name *keystore*. This command prompts for a keystore password, for information that will be used to generate the key, and for a password for the key itself. If the password on the key is different from the password on the keystore, the values KeystoreAlias and KeystoreAliasPassword will be required to retrieve the key.

   ```
   keytool -genkey -v -alias <alias> -keystore <keystore>
   ```

2. Generate a certificate signing request. The following command-line example shows how to generate a certificate signing request for the key entry named *alias* in the keystore named *keystore*, which is then stored in the file named *csr_file*. This file can be sent to a CA to be signed.

   ```
   keytool -certreq -v -alias <alias> -keystore <keystore> -file <csr_file>
   ```

3. Import the CA’s certificate into the keystore. The keytool checks the chain of trust on the user’s certificate upon import. If the certificate was signed by a CA that is not well-known and the keytool knows nothing about the CA, the certificate is rejected. Therefore any certificate from a CA that is not well-known must first be imported into the keystore to permit the user’s certificate to successfully be imported in the next step. The following command-line example shows how to import a certificate in a file named *cert_file* into the keystore named *keystore*:

   ```
   keytool -importkeystore -v -alias <alias> -keystore <keystore> -file <cert_file>
   ```
keytool -import -v -alias <ca_alias> -keystore <keystore> -file <cert_file>

4. Import the signed certificate back into the keystore. Once the certificate signing request has been received by a CA and the signed certificate is sent back from the CA, the certificate can be read into the keystore entry identified by alias. The following command-line example shows how to import the signed certificate.

keytool -import -v -alias <alias> -keystore <keystore_name> -file <csr_file>

5. Check that everything is in the keystore.

keytool -list -v -keystore <keystore_name>

Creating a Truststore

This section describes the basic process for generating a truststore. A truststore is necessary when an SSL provider uses keys that have not been signed by a well-known Certification Authority. A truststore contains only public certificates that have been verified by the person managing the truststore (the trust manager) for the content server. You will need to determine the specific requirements and name for the truststore you want to create. You can store a truststore file wherever you want, because the sslconfig.hda file contains a full path for a TruststoreFile config setting. However, it is recommended that a truststore file is stored in the <stellent_dir>/data/providers/<provider_name> directory (next to the provider.hda and sslconfig.hda files) or in the <stellent_dir>/config/ directory.

For detailed information on how to use the keytool utility to generate a truststore refer to the document titled keytool Key and Certification Management Tool, available online from Sun at www.sun.com.

To use keytool you must have the utility in your path when you enter the command.

To create a truststore enter the following command:

keytool -import -v -alias <alias> -keystore <keystore> -file <cert_files>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;alias&gt;</td>
<td>The alias name for the key.</td>
</tr>
<tr>
<td>&lt;keystore&gt;</td>
<td>The name of the keystore.</td>
</tr>
<tr>
<td>&lt;cert_files&gt;</td>
<td>The path to the Certification Authority’s certificate.</td>
</tr>
</tbody>
</table>
MANAGING SSL AND KEEPALIVE PROVIDERS

The following tasks are involved in managing SSL and keepalive security providers:

- Adding an Incoming Security Provider (page 3-9)
- Adding an Outgoing Security Provider (page 3-10)

For detailed information on managing providers for the content server and how to edit or delete providers refer to *Stellent Content Server Managing System Settings and Processes*.

Adding an Incoming Security Provider

To add an incoming security provider, follow these steps:

1. Display the Providers (page 3-11).
2. In the Create a New provider table, click Add in the Action column for the incoming security provider type.
   
   The incoming security provider page is displayed.
3. Complete the following fields:
   
   **Required fields**
   
   - Provider Name
   - Provider Description
   - Provider Class (provided)
   - Server Port

   **Optional fields**
   
   - Connection Class (provided)
   - Configuration Class
   - Server Thread Class (provided)

   **Optional check boxes (sslincoming provider only)**
   
   - Request Client Authentication
   - Require Client Authentication
4. Click Add.
   
   The Providers page is displayed, with the new provider added to the Providers table.
5. Restart the content server.
Adding an Outgoing Security Provider

To add an outgoing security provider, follow these steps:

1. Display the Providers (page 3-11).

2. In the Create a New provider table, click **Add** in the Action column for the outgoing security provider type.

   The outgoing security provider page is displayed.

3. Complete the following fields:

   **Required fields**
   - Provider Name
   - Provider Description
   - Provider Class (provided)
   - Server Host Name (provided)
   - Server Port
   - Instance Name
   - Relative Web Root

   **Optional fields**
   - Connection Class (provided)
   - Configuration Class
   - Request Class (provided)
   - Number of Connections (provided)
   - HTTP Server Address
   - Proxied (check box)
   - Notify Target (check box)
   - Users (check box)
   - Released Documents (check box)
   - Enterprise Searchable (check box)
   - Required Roles
   - Account Filter

4. Click **Add**.

   The Providers page is displayed, with the new provider added to the Providers table.
5. Restart the content server.

SECURITY PROVIDER INTERFACE PAGES

The Security Providers component enables the following provider interface screens to manage security providers:

- Add keepaliveincoming Provider Page (page 3-13)
- Add keepaliveoutgoing Provider Page (page 3-14)
- Add sslincoming Provider Page (page 3-17)
- Add ssloutgoing Provider Page (page 3-19)

Providers Page

When the Security Providers component has been installed and enabled, the Providers page enables administrators to create SSL and keepalive providers. To access this page click the Providers link on the Administration menu for the Content Server.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Description</th>
<th>Type</th>
<th>Connection State</th>
<th>Last Activity Date</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemDatabase</td>
<td>SystemD</td>
<td>database</td>
<td>16 out of 16 connections are good</td>
<td>3/27/20 3:29 PM</td>
<td>Info Text</td>
</tr>
<tr>
<td>SystemServerSocket</td>
<td>SystemS</td>
<td>incoming</td>
<td>good</td>
<td>3/27/20 3:29 PM</td>
<td>Info Text</td>
</tr>
<tr>
<td>DefaultFileStore</td>
<td>DefaultF</td>
<td>Filestore</td>
<td>good</td>
<td></td>
<td>Test</td>
</tr>
</tbody>
</table>

Create a New Provider

<table>
<thead>
<tr>
<th>Provider Type</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>outgoing</td>
<td>Configuring an outgoing provider.</td>
<td>Add</td>
</tr>
<tr>
<td>database</td>
<td>Configuring a database provider.</td>
<td>Add</td>
</tr>
<tr>
<td>incoming</td>
<td>Configuring an incoming provider.</td>
<td>Add</td>
</tr>
<tr>
<td>preview</td>
<td>Configuring a preview provider.</td>
<td>Add</td>
</tr>
<tr>
<td>ldapserver</td>
<td>Configuring an LDAP user provider.</td>
<td>Add</td>
</tr>
<tr>
<td>httpoutgoing</td>
<td>Configuring an HTTP outgoing provider.</td>
<td>Add</td>
</tr>
<tr>
<td>keepaliveincoming</td>
<td>Configuring a keepalive incoming socket provider.</td>
<td>Add</td>
</tr>
<tr>
<td>keepaliveoutgoing</td>
<td>Configuring a keepalive outgoing socket provider.</td>
<td>Add</td>
</tr>
<tr>
<td>sslincoming</td>
<td>Configuring an SSL incoming socket provider.</td>
<td>Add</td>
</tr>
<tr>
<td>ssloutgoing</td>
<td>Configuring an SSL outgoing socket provider.</td>
<td>Add</td>
</tr>
</tbody>
</table>
### Feature | Description
--- | ---
**Providers table** |  
Provider column | The name of the provider that establishes connection to outside entities.  
Description column | A description of the provider.  
Provider Type column | The type of provider. For example, *incoming*, or *database*.  
Connection State column | Possible states are:  
• misconfigured  
• good  
• down  
• requires restart  
Last Activity Date column | The last date and time that the provider was active.  
Actions column | • The Info link displays the Provider Information Page for the provider.  
• The Test link refreshes the Connection State and Last Activity columns for the provider.  
**Create a New Provider table** |  
Provider Type column | The type of provider.  
Description column | A description of the provider type.  
Action column | Clicking an Add button displays the Add/Edit Provider page for that type of provider.
Add keepaliveincoming Provider Page

The Add Incoming Provider page for the keepalive function is used to create a keepalive socket incoming provider. To access this page click the Providers link on the Administration menu for the Content Server, then click **Add** in the Action column for the keepaliveincoming provider type.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider Name field*</td>
<td>The name of the provider.</td>
</tr>
<tr>
<td>Provider Description field*</td>
<td>A description of the provider.</td>
</tr>
<tr>
<td>Provider Class field*</td>
<td>The name of the Java class for the provider. For example: <code>idc.provider.ExtendedSocketIncomingProvider</code></td>
</tr>
<tr>
<td>Connection Class field</td>
<td>The name of the Java class that implements the provider connection. For example: <code>idc.provider.KeepaliveSocketIncomingConnection</code></td>
</tr>
<tr>
<td>Configuration Class field</td>
<td>The name of a Java class that performs some extra configuration. This class is very useful for database providers, where the connection classes are already providers.</td>
</tr>
</tbody>
</table>
Using Security Providers

The Add Outgoing Provider page for the keepalive function enables administrators to create a keepalive socket outgoing provider. To access this page, select the Providers link from the Administration menu for the Content Server, then select Add in the Action column for the keepaliveoutgoing provider type.

The following two images show this page.

### Add keepaliveoutgoing Provider Page

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Thread field</td>
<td>The name of the server thread for incoming connections. For example: <code>idc.provider.KeepaliveIdcServerThread</code></td>
</tr>
<tr>
<td>Server Port field*</td>
<td>The port the provider listens on for incoming connections. For example, the incoming system provider listens on port 4444 by default.</td>
</tr>
<tr>
<td>Add button</td>
<td>Saves the provider information.</td>
</tr>
<tr>
<td>Reset button</td>
<td>Resets the provider information to the last saved values.</td>
</tr>
</tbody>
</table>

* Required metadata fields.
### Using Security Providers

#### Feature Administration

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider Name field*</td>
<td>The name of the provider.</td>
</tr>
<tr>
<td>Provider Description field*</td>
<td>A description of the provider.</td>
</tr>
<tr>
<td>Provider Class field*</td>
<td>The name of the Java class for the provider. For example: \textit{idc.provider.KeepaliveSocketOutgoingProvider}</td>
</tr>
<tr>
<td>Connection Class field</td>
<td>The name of the Java class that implements the provider connection. For example: \textit{idc.provider.KeepaliveSocketOutgoingConnection}</td>
</tr>
</tbody>
</table>
### Using Security Providers

<table>
<thead>
<tr>
<th><strong>Feature</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Class field</td>
<td>The name of a Java class that performs some extra configuration.</td>
</tr>
<tr>
<td>Request Class field</td>
<td>The name of the Java class that implements the server request. For example: <em>idc.provider.KeepaliveServerRequest</em></td>
</tr>
<tr>
<td>Number of Connections field</td>
<td>The maximum number of connections. For example, 3.</td>
</tr>
<tr>
<td>Server Host Name field*</td>
<td>The server host name of the other content server instance. For example, <em>localhost</em>.</td>
</tr>
<tr>
<td>HTTP Server Address field</td>
<td>The HTTP address of the other content server instance.</td>
</tr>
<tr>
<td>Server Port field*</td>
<td>The port on which the provider communicates with the other content server.</td>
</tr>
<tr>
<td>Instance Name field*</td>
<td>The instance name of the other content server instance.</td>
</tr>
<tr>
<td>Relative Web Root field*</td>
<td>The relative web root of the other content server instance.</td>
</tr>
<tr>
<td>Proxied check box</td>
<td>Enable this option if the provider is connecting to a content server that will be controlled by the current instance.</td>
</tr>
<tr>
<td>Notify Target check box</td>
<td>Enable this option if the provider is connecting to a content server that is acting as a controlling instance, and you want this content server to notify the controlling instance when user information and/or content item information changes.</td>
</tr>
<tr>
<td>Users check box</td>
<td>Enable this option if you want this content server to notify the controlling instance when user information changes.</td>
</tr>
<tr>
<td>Released Documents check box</td>
<td>Enable this option if you want this content server to notify the controlling instance when content item information changes.</td>
</tr>
</tbody>
</table>
### Using Security Providers

**Add sslincoming Provider Page**

The AddIncoming Provider page for the sslincoming function enables administrators to create an SSL socket incoming provider. To access the page click the Providers link from the Administration menu for the Content Server, then select **Add** from the Action column for the “sslincoming” provider type.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enterprise Searchable check box</strong></td>
<td>Enable this option if you have enabled Enterprise Search and you want this content server instance to be searchable. See the <em>Stellent Enterprise Search Administration and User Guide</em> for more information.</td>
</tr>
<tr>
<td><strong>Required Roles field</strong></td>
<td>Enter roles that have permission to search this content server instance using Enterprise Search. If no roles are entered, all users will have permission.</td>
</tr>
<tr>
<td><strong>Account Filter field</strong></td>
<td>Enter accounts that have permission to search this content server instance using Enterprise Search. If no accounts are entered, all users will have permission.</td>
</tr>
<tr>
<td><strong>Conversion options</strong></td>
<td>Used to indicate if the provider is an Inbound Refinery.</td>
</tr>
<tr>
<td><strong>Add button</strong></td>
<td>Saves the provider information.</td>
</tr>
<tr>
<td><strong>Reset button</strong></td>
<td>Resets the provider information to the last saved values.</td>
</tr>
</tbody>
</table>

* Required metadata fields.
Using Security Providers

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider Name field*</td>
<td>The name of the provider.</td>
</tr>
<tr>
<td>Provider Description field*</td>
<td>A description of the provider.</td>
</tr>
<tr>
<td>Provider Class field*</td>
<td>The name of the Java class for the provider. For example: <code>idc.provider.ssl.SSLSocketIncomingProvider</code></td>
</tr>
<tr>
<td>Connection Class field</td>
<td>The name of the Java class that implements the provider connection. For example: <code>idc.provider.KeepaliveSocketIncomingConnection</code></td>
</tr>
<tr>
<td>Configuration Class field</td>
<td>The name of a Java class that performs some extra configuration. This class is very useful for database providers, where the connection classes are already providers.</td>
</tr>
<tr>
<td>Server Thread field</td>
<td>The name of the server thread for incoming connections. For example: <code>idc.provider.KeepaliveIdcServerThread</code></td>
</tr>
<tr>
<td>Server Port field*</td>
<td>The port the provider listens on for incoming connections. For example, the incoming system provider listens on port 4444 by default.</td>
</tr>
</tbody>
</table>
Add ssloutgoing Provider Page

The Add Outgoing Provider page for the ssloutgoing function enables administrators to create an SSL socket outgoing provider. To access the page click the Providers link from the Administration menu for the Content Server, then select **Add** from the Action column for the “ssloutgoing” provider type.

The following two images show this page.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Client Authentication check box</td>
<td>Enable this option if you want the provider to request client authentication from the incoming connection.</td>
</tr>
<tr>
<td>Require Client Authentication check box</td>
<td>Enable this option if you want the provider to require client authentication from the incoming connection.</td>
</tr>
<tr>
<td>Keystore/Alias/Truststore information</td>
<td>If necessary, enter the keystore password name, the alias, the alias password, and the truststore password.</td>
</tr>
<tr>
<td>Add button</td>
<td>Saves the provider information.</td>
</tr>
<tr>
<td>Reset button</td>
<td>Resets the provider information to the last saved values.</td>
</tr>
</tbody>
</table>

* Required metadata fields.
Using Security Providers

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider Name field*</td>
<td>The name of the provider.</td>
</tr>
<tr>
<td>Provider Description field*</td>
<td>A description of the provider.</td>
</tr>
</tbody>
</table>

* Required fields.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider Class field*</td>
<td>The name of the Java class for the provider. For example:</td>
</tr>
<tr>
<td></td>
<td><code>idc.provider.KeepaliveSocketOutgoingProvider</code></td>
</tr>
<tr>
<td>Connection Class field</td>
<td>The name of the Java class that implements the provider connection. For example:</td>
</tr>
<tr>
<td></td>
<td><code>idc.provider.KeepaliveSocketOutgoingConnection</code></td>
</tr>
<tr>
<td>Configuration Class field</td>
<td>The name of a Java class that performs some extra configuration.</td>
</tr>
<tr>
<td>Request Class field</td>
<td>The name of the Java class that implements the server request. For example:</td>
</tr>
<tr>
<td></td>
<td><code>idc.provider.KeepaliveServerRequest</code></td>
</tr>
<tr>
<td>Number of Connections field</td>
<td>The maximum number of connections. For example, 3.</td>
</tr>
<tr>
<td>Server Host Name field*</td>
<td>The server host name of the other content server instance. For example, <code>localhost</code>.</td>
</tr>
<tr>
<td>HTTP Server Address field</td>
<td>The HTTP address of the other content server instance.</td>
</tr>
<tr>
<td>Server Port field*</td>
<td>The port on which the provider communicates with the other content server.</td>
</tr>
<tr>
<td>Instance Name field*</td>
<td>the instance name of the other content server instance.</td>
</tr>
<tr>
<td>Relative Web Root field*</td>
<td>The relative web root of the other content server instance.</td>
</tr>
<tr>
<td>Keystore/Alias/Truststore information</td>
<td>If necessary, enter the keystore password name, the alias, the alias password, and the truststore password.</td>
</tr>
<tr>
<td>Proxied check box</td>
<td>Enable this option if the provider is connecting to a content server that will be controlled by the current instance.</td>
</tr>
<tr>
<td>Notify Target check box</td>
<td>Enable this option if the provider is connecting to a content server that is acting as a controlling instance, and you want this content server to notify the controlling instance when user information and/or content item information changes.</td>
</tr>
</tbody>
</table>
### Feature

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users check box</td>
<td>Enable this option if you want this content server to notify the controlling instance when user information changes.</td>
</tr>
<tr>
<td>Released Documents check box</td>
<td>Enable this option if you want this content server to notify the controlling instance when content item information changes.</td>
</tr>
<tr>
<td>Enterprise Searchable check box</td>
<td>Enable this option if you have enabled Enterprise Search and you want this content server instance to be searchable. See the <em>Stellent Enterprise Search Administration and User Guide</em> for more information.</td>
</tr>
<tr>
<td>Required Roles field</td>
<td>Enter roles that have permission to search this content server instance using Enterprise Search. If no roles are entered, all users will have permission.</td>
</tr>
<tr>
<td>Account Filter field</td>
<td>Enter accounts that have permission to search this content server instance using Enterprise Search. If no accounts are entered, all users will have permission.</td>
</tr>
<tr>
<td>Conversion options</td>
<td>Used to indicate if the provider is an Inbound Refinery.</td>
</tr>
<tr>
<td>Add button</td>
<td>Saves the provider information.</td>
</tr>
<tr>
<td>Reset button</td>
<td>Resets the provider information to the last saved values.</td>
</tr>
</tbody>
</table>

* Required field
A
Add Incoming Provider page
keepalive, 3-13
sslincoming, 3-17
Add Outgoing Provider page
keepalive, 3-14
ssloutgoing, 3-19

B
backward slashes
convention in document, 1-3

C
Certification Authority, 3-5, 3-8
cOMPONENT file
contents of, 1-2
convention
backward slashes, 1-3
dialog or window paths, 1-3
forward slashes, 1-3
system location, 1-3
user input, 1-3
conventions
used in document, 1-3

D
dialog path
convention in document, 1-3

F
forward slashes
convention in document, 1-3

I
installing the Security Providers component, 2-1

K
keepalive providers
adding keepaliveincoming provider, 3-13
adding keepaliveoutgoing provider, 3-14
additional configuration, 3-5
description of, 3-3
Java classes, 3-3
keystores
description of, 3-5
generating, 3-6
keytool utility, 3-7
Portecle Java tool, 3-7
specifying, 3-6
when to use, 3-6

P
provider.hda file, 3-6

S
security
sources of information about, 3-1
terms, 3-2
security providers
adding incoming, 3-9
adding outgoing, 3-10
interface screens, 3-11
keepalive, 3-3
Index

planning for, 3-3
SSL, 3-4
SecurityProviders component
installing, 2-1
SSL providers
adding sslincoming provider, 3-17
adding ssloutgoing provider, 3-19
additional configuration, 3-5
description of, 3-4
Java classes, 3-4
keystores, 3-5
truststore, 3-5
sslconfig.hda file, 3-6
system location
convention in document, 1-3

T
truststore
creating a, 3-8
description of, 3-5
keytool utility, 3-8
specifying, 3-6
when to use, 3-6

U
user input
convention in document, 1-3

W
window path
convention in document, 1-3