System Administration Guide

Jaguar CTS
Version 3.6.1
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How does the Jaguar naming service work?</td>
<td>47</td>
</tr>
<tr>
<td>Jaguar server initial context</td>
<td>48</td>
</tr>
<tr>
<td>Name binding example</td>
<td>49</td>
</tr>
<tr>
<td>Transient vs. persistent storage</td>
<td>51</td>
</tr>
<tr>
<td>CORBA CosNaming API support</td>
<td>51</td>
</tr>
<tr>
<td>Binding names</td>
<td>52</td>
</tr>
<tr>
<td>Resolving Jaguar objects</td>
<td>52</td>
</tr>
<tr>
<td>Resolving objects using the CosNaming interface</td>
<td>53</td>
</tr>
<tr>
<td>JNDI support</td>
<td>55</td>
</tr>
<tr>
<td>JNDI J2EE features</td>
<td>55</td>
</tr>
<tr>
<td>Configuring the Jaguar naming service</td>
<td>60</td>
</tr>
<tr>
<td>Name binding password security</td>
<td>61</td>
</tr>
<tr>
<td>Using an LDAP server with Jaguar</td>
<td>61</td>
</tr>
<tr>
<td>LDAP object schema and Jaguar objects</td>
<td>62</td>
</tr>
<tr>
<td>Storing Jaguar object bindings on an LDAP server</td>
<td>62</td>
</tr>
<tr>
<td><strong>CHAPTER 3</strong> Jaguar Clusters and Synchronization</td>
<td>63</td>
</tr>
<tr>
<td>Cluster overview</td>
<td>63</td>
</tr>
<tr>
<td>Cluster servers</td>
<td>65</td>
</tr>
<tr>
<td>Configuring a cluster</td>
<td>66</td>
</tr>
<tr>
<td>Heartbeat detection</td>
<td>70</td>
</tr>
<tr>
<td>Cluster start-up options</td>
<td>72</td>
</tr>
<tr>
<td>Synchronization</td>
<td>73</td>
</tr>
<tr>
<td>Component synchronization</td>
<td>75</td>
</tr>
<tr>
<td>Package synchronization</td>
<td>76</td>
</tr>
<tr>
<td>Servlet synchronization</td>
<td>76</td>
</tr>
<tr>
<td>Application synchronization</td>
<td>76</td>
</tr>
<tr>
<td>Web application synchronization</td>
<td>77</td>
</tr>
<tr>
<td><strong>CHAPTER 4</strong> Load Balancing, Failover, and Component Availability</td>
<td>81</td>
</tr>
<tr>
<td>Load balancing overview</td>
<td>82</td>
</tr>
<tr>
<td>Load metrics</td>
<td>82</td>
</tr>
<tr>
<td>Load distribution policies</td>
<td>83</td>
</tr>
<tr>
<td>Interoperable object references</td>
<td>84</td>
</tr>
<tr>
<td>Configuring load balancing</td>
<td>85</td>
</tr>
<tr>
<td>Component deployment</td>
<td>86</td>
</tr>
<tr>
<td>High availability</td>
<td>87</td>
</tr>
<tr>
<td>Automatic failover</td>
<td>88</td>
</tr>
<tr>
<td>Component guidelines</td>
<td>89</td>
</tr>
<tr>
<td>Implementing Sybase Failover for high availability systems</td>
<td>91</td>
</tr>
<tr>
<td>JDBC 1.0 and 2.0</td>
<td>91</td>
</tr>
<tr>
<td>JDBC 2.0 extn/JTA drivers</td>
<td>92</td>
</tr>
<tr>
<td>Troubleshooting the database connection</td>
<td>93</td>
</tr>
</tbody>
</table>
CHAPTER 5  Security Configuration .................................................................  95
  Public-key cryptography .................................................................  96
  Public-key certificates ................................................................  96
  SSL, HTTPS, and IIOPS .............................................................  97
  Jaguar security management overview ......................................  98
  Entrust integration ....................................................................... 99
    Both client and Jaguar server use non-Entrust certificates .... 99
    Entrust client and non-Entrust Jaguar server (and vice versa) 100
    Both client and Jaguar server use Entrust certificates ........ 100
  Using Netscape to manage certificates on the client ............... 100
    Installing Sybase PKCS #11 into Netscape 4.0x ....................... 101
    Obtaining a key pair and certificate ........................................ 102
    SSL certificate information in servlets .................................... 103
  Security Manager tasks ................................................................ 104
    Security Manager management ............................................... 104
    Test CA management ............................................................. 105
    Key management .................................................................. 111
    Certificate management ....................................................... 112
  Security profiles ....................................................................... 121
    Configuring security profiles ................................................... 125
  Listeners ..................................................................................... 129
    Preconfigured listeners ............................................................ 129
    Configuring listeners .............................................................. 131
  Jaguar quality of protection ....................................................... 133
    Usage scenarios .................................................................... 133
  Administration password and OS authentication ...................... 137
  Roles ............................................................................................ 139
    Assigning users and groups to roles ...................................... 141
    Admin role in Jaguar .............................................................. 144
About This Book

<table>
<thead>
<tr>
<th>Subject</th>
<th>This book contains information about configuring Jaguar Component Transaction Servers (Jaguar CTS™).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audience</td>
<td>This book is for anyone responsible for configuring the Jaguar CTS runtime environment, establishing security, or for creating and deploying packages and components on a Jaguar server.</td>
</tr>
<tr>
<td>How to use this book</td>
<td>Chapter 1, “Jaguar Configuration” contains information about configuring the Jaguar runtime environment, including:</td>
</tr>
</tbody>
</table>

  - Creating Jaguar servers
  - Setting server properties
  - Setting HTTP properties
  - Configuring connection caches
  - Configuring the message service

Chapter 2, “Jaguar Naming Services” contains information about using Jaguar naming services to locate objects—such as packages, components, and servers—anywhere on the network.

Chapter 3, “Jaguar Clusters and Synchronization” contains information about creating a cluster of Jaguar servers, which provides high availability for Jaguar services and components, and synchronizing repositories from a primary server within a cluster to other clustered servers, which keeps all of the servers within the cluster up to date.

Chapter 4, “Load Balancing, Failover, and Component Availability” contains information about how to load balance between a cluster of Jaguar servers and how to configure and implement component failover.

Chapter 5, “Security Configuration” contains information about configuring SSL features for Jaguar, including:

  - Generating client/server certificates
  - Assigning certificates to security profiles
  - Assigning security profiles to listeners
It also includes information about:

- Changing the administration password
- Using operating system authentication
- Assigning roles to packages and components

Related documents

The *Enterprise Application Server Installation Guide Supplement* for your platform contains installation instructions for Solaris and Windows NT. The installation CD includes a *README.txt* file that contains the latest information about Jaguar.

The *Jaguar CTS Getting Started* guide contains tutorials that help you become more familiar with Jaguar features. It also contains instructions for starting Jaguar servers and Jaguar Manager. *New Features* contains information about new features in Jaguar.

The *Jaguar CTS Programmer’s Guide* contains detailed information about C/C++, Java, and ActiveX components, Java stub classes and stub generation, session management, event handlers, and connection management as well as Jaguar’s programming models. It provides information on Jaguar-supported datatypes, result-set handling, and compiling dynamic link libraries (DLLs).

The *Jaguar CTS API Reference* contains reference pages for the Java classes and interfaces and C routines used by Jaguar. It is available in HTML format in the */html/docs* directory of your Jaguar installation.

The Sybase Web page provides white papers and other product information. See:


Other sources of information

Use the Sybase Online Books CD and the Sybase Product Manuals Web site to learn more about your product:

- Online Books CD contains product manuals and technical documents and is included with your software. The DynaText browser (included on the Online Books CD) allows you to access technical information about your product in an easy-to-use format.

See the *Installation Guide* in your documentation package for instructions on installing and starting Online Books.
About this book

• Sybase Product Manuals Web site is an HTML version of the Online Books CD that you can access using a standard Web browser. In addition, you’ll find links to the Technical Documents Web site (formerly known as Tech Info Library), the Solved Cases page, and Sybase/Powersoft newsgroups.

To access the Sybase Product Manuals Web site, go to the Sybase Support Services site at http://www.sybase.com/support/manuals and select a link under the Product Manuals heading.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ For the latest information on product certifications and/or the EBF Rollups:

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2 In the Browse section, click on What’s Hot.
3 Select links to Certification Reports and EBF Rollups, as well as links to Technical Newsletters, online manuals, and so on.

❖ If you are a registered SupportPlus user:

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2 In the Browse section, click on What’s Hot.
3 Click on EBF Rollups. You can research EBFs using Technical Documents, and you can download EBFs using Electronic Software Distribution (ESD).
4 Follow the instructions associated with the SupportPlusSM Online Services entries.

❖ If you are not a registered SupportPlus user, and you want to become one:

You can register by following the instructions on the Web. To use SupportPlus, you need:

1 A Web browser that supports the Secure Sockets Layer (SSL), such as Netscape Navigator 1.2 or later.
2 An active support license.
3 A named technical support contact.
4 Your user ID and password.

❖ Whether or not you are a registered SupportPlus user:

You may use Sybase’s Technical Documents. Certification Reports are among the features documented at this site.

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/

2 In the Browse section, click on What’s Hot.

3 Click on the topic that interests you.

Conventions

Throughout the rest of this manual:

• Jaguar refers to Jaguar Component Transaction Server.

• EAServer refers to Enterprise Application Server.

The syntax conventions used in this manual are:

<table>
<thead>
<tr>
<th>Key</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>commands and methods</td>
<td>Command names, utility names, utility flags, methods, and other keywords are in lowercase Arial font.</td>
</tr>
<tr>
<td>variable, package, or component</td>
<td>Variables, or words that stand for values that you fill in, file, package, and component names are in italics.</td>
</tr>
<tr>
<td>menus and menu options</td>
<td>The vertical bar shows you how to navigate menu selections. For example, selecting Program Files</td>
</tr>
<tr>
<td>package 1</td>
<td>Information that you enter in Jaguar Manager and programming code are shown in Courier font.</td>
</tr>
</tbody>
</table>

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.
CHAPTER 1

Jaguar Configuration

This chapter describes basic configuration tasks that you can perform to customize your installation, such as creating new servers, changing server properties, and defining new connection caches.

The Jaguar runtime environment is preconfigured. With minimum setup, you can have a fully functioning transaction server. Although the default settings are usually sufficient, Jaguar provides you with the flexibility to customize your server environment when necessary.

You can perform all configuration tasks using the Jaguar Manager and Security Manager.

### Related chapters

For information on configuring Jaguar naming services, see Chapter 2, “Jaguar Naming Services”. For information on configuring all aspects of Jaguar security, including establishing and changing listeners, native SSL support, using digital certificates, roles, Web application security, and so on, see Chapter 5, “Security Configuration”.

### Creating or deleting a server

- **To create a Jaguar server:**
  1. Double-click the Jaguar Manager icon.
Configuring a Jaguar server

2  Highlight the Servers icon.
3  Select File | New Server.
4  Enter a name for the new server. Server names must be one word, and can be up to 255 characters long. Click Create New Server.
5  Complete the server configuration. For information see “Configuring a Jaguar server” on page 2.
6  For each new server you create, add an HTTP, TDS, and IIOP listener. See “Preconfigured listeners” on page 129 for more information.
7  Start the server.

❖ To delete an existing Jaguar server:
   1  Double-click the Jaguar Manager icon.
   2  Double-click the Servers icon.
   3  Highlight the server you want to delete.
   4  Select File | Delete Server.

Note  You cannot delete the server to which the Jaguar Manager is connected. At least one server must be defined in your Jaguar installation.

Configuring a Jaguar server

To configure or modify the properties of an individual Jaguar server:

1  From within Jaguar Manager, display the list of installed servers by double-clicking the Servers folder.
2  Highlight the server you want to configure.
3  Select File | Server Properties. You see the Server Properties window, which contains these tabs:
   •  General – define general individual server properties.
   •  HTTP Config – determine browser accessibility.
   •  Transactions – determine the transaction coordinator for components that participate in Jaguar transactions.
• **Security** – see “Administration password and OS authentication” on page 137 for security options accessible from this tab.

• **Resources** – define the maximum number of client connections.

• **Log/Trace** – set logging and trace options.

• **Handlers** – define individual server event handlers.

• **Naming Service** – set the Jaguar server’s naming service options. See Chapter 2, “Jaguar Naming Services” for additional information about naming services.

• **Servlets** – disable servlet execution in Jaguar and configure additional properties to control the execution of servlets.

• **Java VM** – set the version of the Java virtual machine.

• **Dynamo** – enable hosting of PowerDynamo™ Web sites in Jaguar.

• **Hot standby** – Enable hot standby and define the Master and backup servers.

• **All Properties** – edit server property settings in their raw format, that is, as they are stored in the configuration repository.

If you modify any property, click OK in the Server Properties sheet to save your changes, or click Cancel to disregard the changes.

When you modify server properties you must refresh the server for the changes to take effect. To refresh the server, highlight the server icon and select File | Refresh.

### General

Table 1-1 describes the general properties that you can configure for individual servers.
Table 1-1: Server general properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Enter a description of the server, up to 255 characters in length.</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Codeset</td>
<td>Specify the character set used by the server.</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Classpath</td>
<td>Displays the contents of the CLASSPATH environment variable for the server that you are connected to. This setting specifies the directories from which Java class files can be loaded. It is defined by the start-up script when you start the server. The CLASSPATH does not display for servers that you are not connected to.</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

HTTP Config

Clients can access Jaguar and retrieve HTML pages using a Web browser. You can customize certain aspects of your server’s HTTP behavior by modifying the HTTP configuration properties listed in Table 1-2.

Table 1-2: HTTP properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default value</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Name</td>
<td>Domain name in .company.xxx format.</td>
<td>N/A</td>
<td>.sybase.com</td>
</tr>
<tr>
<td>Document Root</td>
<td>The path to the directory where documents are served.</td>
<td>$JAGUAR/html</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(UNIX)</td>
<td>%JAGUAR%html</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(NT)</td>
<td>%work\WWW\</td>
<td></td>
</tr>
<tr>
<td>Default HTML File</td>
<td>The default HTML file served.</td>
<td>index.html</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/work/WWW/index.html (UNIX)</td>
<td>%work\WWW\index.html (NT)</td>
</tr>
</tbody>
</table>
HTTP logging and statistics

Jaguar maintains two files that allow you to monitor HTTP events.

Request log information is recorded in `httprequest.log`. HTTP statistics are logged in the file `httpstat.dat`. Both files are located by default in the Jaguar `bin` subdirectory (or `devbin` if you are running the debug server version).

The request log records information about each HTTP request. If you define additional servers, the name of the `httprequest.log` file is prepended with the server name. For example, if you create a server named Test_server, messages for that server are directed to the `Test_serverhttprequest.log` file.

The statistics log records the total number of hits on the server and the total number of hits per page.

For information on viewing these files, see “File Viewer” on page 33.
Transactions

This section describes the transaction coordinator models that are available. All components installed in one Jaguar server share the same transaction coordinator.

Jaguar transaction coordinator models are:

- **Shared connection** – for NT or UNIX users this option, which is the default, is a “pseudo-coordinator” that is built into Jaguar. All components participating in a transaction share a single connection. To use this model, all your application data must reside on one data server, and all components that participate in a transaction must use a connection with the same user name and password.

- **Microsoft Distributed Transaction Coordinator (DTC)** – for NT users only, DTC uses two-phase commit to coordinate transactions among multiple databases. DTC is available on Windows NT as part of Microsoft SQL Server 6.5.

- **Object Transaction Service X/Open Architecture (OTS/XA) Transactions** – for NT or UNIX users, this option is the Transarc Encina® transaction coordinator that is built into Jaguar. The Encina transaction coordinator uses two-phase commit to coordinate transactions among multiple databases.

**Note** To enable the OTS/XA transaction coordinator, you must create the transaction coordinator log file as follows:

- For Solaris, execute:
  
  ```bash
  echo x | dd seek=8k of=JagOTSLog.dev
  ```

- For NT, execute:
  
  ```cmd
  Filevol JagOTSLog.dev 4000K
  ```

  If you do not create a log file, the Jaguar server will not start up.

To set the transaction coordinator for your Jaguar server, select the transaction model from the server’s Transactions tab in the Server Properties window.

For detailed information about components and transactions, see Chapter 3, “Understanding Transactions and Component Lifecycles” in the *Jaguar CTS Programmer’s Guide*. 
Resources

The Resources tab allows you to limit the number of concurrent client sessions and contains configurable properties used by Open Server applications. Table 1-3 describes the server resource properties.

### Table 1-3: Server resource properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Number Client Sessions</td>
<td>The maximum number of concurrent client sessions supported by Jaguar. This does not include HTTP sessions, which are controlled by the maximum thread property described in “HTTP Config” on page 4. Modify this variable as needed to balance system resources versus session availability.</td>
<td>30</td>
</tr>
<tr>
<td>Thread Stack Size (Available on UNIX platforms only)</td>
<td>The stack size for server threads, specified in bytes as a decimal number.</td>
<td></td>
</tr>
<tr>
<td>Message Pool Size</td>
<td>The number of messages available to an Open Server application at runtime.</td>
<td></td>
</tr>
<tr>
<td>Message Queue Size</td>
<td>The number of message queues available to an Open Server application.</td>
<td></td>
</tr>
<tr>
<td>Network Buffer Size</td>
<td>The maximum size of the network I/O buffer to be used by client connections.</td>
<td></td>
</tr>
</tbody>
</table>

Warning! If you increase this value, you must also increase the maximum number of threads.

See “Configuring server stack size” on page 18 for information on setting this property.

These properties are for Open Server applications. See your Open Server documentation for additional information.
Log/Trace

Tracing provides information about activities carried out by your application. Trace output is sent to the Jaguar log file. To establish the level of detail for logging and tracing, select the Log/Trace tab. Table 1-4 describes the logging and trace properties.

For information on viewing the log file, see “File Viewer” on page 33.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Number of Threads</td>
<td>The maximum number of connection threads, including HTTP and IIOP connections. Set this value equal to, or greater than, the sum of the maximum number of HTTP connections and the maximum number of client sessions. See “HTTP Config” on page 4 for information on the HTTP connections value.</td>
<td>50</td>
</tr>
</tbody>
</table>
Chapter 1    Jaguar Configuration

Table 1-4: Debug/Trace properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log File Name</td>
<td>The name of the Jaguar log file. This file defaults to srv.log in the Jaguar bin subdirectory. The srv.log logs a wide range of information and is helpful in isolating problems. You can create the log file in an alternate directory by prefixing a full path to the file name you enter. If you do not enter a full path, the file is created in the Jaguar bin subdirectory. You cannot use environment variables when specifying a full path.</td>
</tr>
<tr>
<td>Log File Size (Bytes)</td>
<td>The size, in bytes, to which the log file grows before it is truncated.</td>
</tr>
<tr>
<td>Truncate Log on Startup</td>
<td>When this flag is set, the log truncates every time the server is restarted. Keep in mind that if the server crashes and this flag is set, you will lose the log file and the information it contains.</td>
</tr>
<tr>
<td>Trace Attentions</td>
<td>If set, traces attentions received or acknowledged by Jaguar.</td>
</tr>
<tr>
<td>Trace Network Driver APIs</td>
<td>If set, traces Net-Lib driver requests.</td>
</tr>
<tr>
<td>Trace Network Driver Requests</td>
<td>If set, traces network layer protocol requests.</td>
</tr>
<tr>
<td>Trace Protocol Data</td>
<td>If set, traces TDS packet content (the actual TDS traffic between a client and Jaguar) in hexadecimal and ASCII format.</td>
</tr>
<tr>
<td>Protocol Headers</td>
<td>If set, traces TDS protocol packet header information, such as packet type and length.</td>
</tr>
<tr>
<td>Trace Servlets</td>
<td>If set, traces the execution of Jaguar’s servlet execution engine.</td>
</tr>
</tbody>
</table>

Handlers

Jaguar supports event-based programming in addition to component-based programming. In the event-based programming model, the application programmer creates custom event handlers that extend the way Jaguar responds to a variety of events. Some of these events are triggered by client requests, while others originate in Jaguar.

For example, if you have coded a connect event handler, here is what happens:

1. The client sends a request to connect to Jaguar.
2. Jaguar receives the request, which generates a connect event.
3. Jaguar executes the connect event handler code.
To specify an event handler:

1. Select the Handlers tab.
2. Enter the DLL or shared library name and the function name of the specific event handler being called, separated by a colon.

The following example illustrates an entry for a connect event handler:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris, AIX, Digital UNIX, and LINUX</td>
<td>libsamp.so:debug_connect</td>
</tr>
<tr>
<td>HP-UX</td>
<td>libsamp.sl:debug_connect</td>
</tr>
<tr>
<td>NT</td>
<td>libsamp.dll:debug_connect</td>
</tr>
</tbody>
</table>

where *libsamp* is the DLL or shared library name and *debug_connect* is the function called whenever a connect event handler is called.

Table 1-5 summarizes the types of event handlers that you can install. For information on coding event handlers, see the *Jaguar CTS Programmer’s Guide*. 
Table 1.5: Individual server event handlers

<table>
<thead>
<tr>
<th>Event handler</th>
<th>Called</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect</td>
<td>Each time a client connects to Jaguar.</td>
</tr>
<tr>
<td>Disconnect</td>
<td>When the client disconnects from Jaguar.</td>
</tr>
<tr>
<td>Error</td>
<td>When a Jaguar server processing error occurs.</td>
</tr>
<tr>
<td>Initialization</td>
<td>Before starting Jaguar.</td>
</tr>
<tr>
<td>Start</td>
<td>When a request to start the Jaguar server is made.</td>
</tr>
<tr>
<td>Stop</td>
<td>When a request to stop the Jaguar server is made.</td>
</tr>
<tr>
<td>Language</td>
<td>When a client sends a language request, such as a SQL statement.</td>
</tr>
<tr>
<td>RPC</td>
<td>When a client issues a remote procedure call.</td>
</tr>
<tr>
<td>Attention</td>
<td>When an attention has been received. An attention is an immediate event; Jaguar services the attention as soon as it occurs, rather than adding it to the client’s event queue.</td>
</tr>
<tr>
<td>Cursor</td>
<td>When a client sends a cursor request.</td>
</tr>
<tr>
<td>Dynamic</td>
<td>When a client sends a dynamic SQL request.</td>
</tr>
<tr>
<td>Message</td>
<td>When the client sends a message.</td>
</tr>
<tr>
<td>Option</td>
<td>When a client sends an option command.</td>
</tr>
<tr>
<td>Bulk</td>
<td>When a client issues a bulk copy request.</td>
</tr>
</tbody>
</table>

Naming Service

Select the Naming Service tab on the Server Properties window to set the Jaguar server’s naming service options. You can use this property sheet to configure a Jaguar server to be a name server, or to point to another Jaguar server as its name server.

Note You can also set the bindpassword server property to enable password protection for name binding on a Jaguar name server. For more information, see “Name binding password security” on page 61.

For general information about naming services, see Chapter 2, “Jaguar Naming Services”.

Initial Context – enter the Jaguar server’s default name context. The name server binds any object implementations on the server to the server’s initial name context.
Some text on configuring a Jaguar server.

If you use a Jaguar server as a name server, the name context can be a compound name with each organization level separated with a forward slash ("/"); for example, /us/sybase/finance.

If you use an external LDAP server to provide persistent storage, the initial context must match the schema used by the LDAP server. For example, c=us,o=sybase,ou=finance.

**Naming server options**

Use these options to specify whether the Jaguar server is also a name server, or whether it uses another Jaguar server as its name server.

- Click Enable as a Name Server to configure the Jaguar server as a name server. If you select this option, you can then also set the Name Server Strategy options described below.
- If the Jaguar server uses another Jaguar server as its name server, unselect Enable as a Name Server. Enter the URL for the Jaguar server acting as the name server; for example, iiop://localhost:9000.

**Naming server strategy**

If you enabled the Jaguar server to be a name server, indicate whether the server provides transient or persistent object name storage. By itself, a Jaguar name server provides transient storage. However, you can add persistent storage capabilities to Jaguar by using an external naming service, such as an LDAP name server.

If you enable persistent storage, enter the following information:

- The URL of the LDAP name server
- A manager DN (distinguished name) for the LDAP server
- The manager DN password

The manager DN provides exclusive access to all objects in the LDAP server database in order to bind and update the objects on the name server. The manager DN and its password are part of the LDAP server configuration properties, set by the server administrator. See your LDAP server documentation for complete information.
Chapter 1    Jaguar Configuration

Servlets

On the Servlet tab in the Server Properties window, you can disable servlet execution in a Jaguar server and configure additional properties to control the execution of servlets.


Java VM

You can choose from several Java virtual machines (VMs) to run with your Jaguar server.

To support multiple Java VMs, a batch file for each supported Java VM ships with your Jaguar installation. For information on running Jaguar with the various Java VMs see Jaguar CTS Getting Started.

You can also set the JAGUAR_JAVAVM variable in Jaguar Manager. This variable is referenced only when starting a Jaguar server to determine which Java VM runs with Jaguar and is ignored if you run one of the batch files listed in Table 1-6 since those files already have a JAGUAR_JAVAVM variable set. If you start a Jaguar server with a batch file that does not have the JAGUAR_JAVAVM variable set then this setting determines which Java VM runs with Jaguar.

To set the JAGUAR_JAVAVM variable from Jaguar Manager:

1. Highlight the server for which you are setting the Java VM.
3. Select the Java VM tab.
4. To change from the default of Java Software JDK 1.1.x, where x is the latest supported version, select the Java VM you want and click OK.

To see which Java VM your Jaguar server is currently using, view the com.sybase.jaguar.server.java.library property from the All Properties tab.
Configuring a Jaguar server

Table 1-6: Java VM options

<table>
<thead>
<tr>
<th>Java VM</th>
<th>JAGUAR_JAVAVM</th>
<th>Batch file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Software JDK 1.1.x</td>
<td>jdk11x</td>
<td>serverstart.bat (NT)</td>
</tr>
<tr>
<td>(The default JDK</td>
<td></td>
<td>srvstart (UNIX)</td>
</tr>
<tr>
<td>included with Jaguar)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java Software JDK 1.2</td>
<td>jdk12</td>
<td>serverstart_jdk12.bat (NT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>srvstart_jdk12 (UNIX)</td>
</tr>
<tr>
<td>Microsoft SDK 3.2</td>
<td>msvm</td>
<td>serverstart_msvm.bat (NT)</td>
</tr>
</tbody>
</table>

Dynamo

This section discusses how to configure Jaguar to host your Dynamo Web sites and provide access to those sites from a browser. Access to Dynamo Web sites is disabled by default.

❖ To host Dynamo Web sites in Jaguar:

1. Install Dynamo version 3.6 or greater and Jaguar 3.6 or greater on your NT or UNIX machine.

2. Update your Jaguar server's environment to include Dynamo DLL and class files. Depending on your platform, perform the following:

   On NT, depending on the virtual machine you are using, modify the %JAGUAR%\bin\serverstart.bat file (JDK 1.1.x VM), %JAGUAR%\bin\serverstart_jdk12.bat file (JDK 1.2), or %JAGUAR%\bin\serverstart_msvm.bat file (Microsoft SDK 3.2):

   - Add %PDYNAMO%\win32 to your PATH.
   - Add %PDYNAMO%\class03 to your CLASSPATH.

   where %PDYNAMO% is the root of your Dynamo installation. You do not need to define a %PDYNAMO% environment variable.

   On UNIX, modify your $JAGUAR/bin/SETUP.sh (Bourne shell) or $JAGUAR/bin/SETUP.csh (C shell) file:

   - Define a PDYNAMO environment variable and set it to the root of your Dynamo installation. This environment variable is required on UNIX machines.
   - Define a JAGUARCLASSES environment variable and set it to your Jaguar client zip file, $JAGUAR/client/jagclient.zip.
   - Add $PDYNAMO/lib to your LD_LIBRARY_PATH.
3 Start your Jaguar server and connect to it from Jaguar Manager. To enable Dynamo support:
   1 From Jaguar Manager, open the Servers folder.
   2 Highlight the Jaguar server.
   3 Select File | Server Properties.
   4 Select the Dynamo tab.
   5 Click Enable Dynamo Execution.
   6 If you have both a Dynamo mapping and a servlet alias with the same URL, select Dynamo or Servlets from the Priority of Jaguar HTTP Services list. This determines whether the Dynamo Web site or servlet is served to the client.
   7 Click OK.

4 Configure your machine so that your Web sites can connect to and retrieve information from databases that they use. For example, on NT, if you load a Web site that accesses data from a SQL Anywhere database, you must include the SQL Anywhere DLLs in your PATH and set up the ODBC data source properly. See your database software instructions and the PowerDynamo User's Guide for detailed information.

---

Warning! If you have a Netscape Web server installed on your machine, Dynamo loads a Netscape version of ns-httpd30.dll instead of a Jaguar version of the same DLL. Rename (but do not delete) Netscape's version of this DLL so that Dynamo loads the Jaguar version instead.

You can now access a Dynamo Web site by entering Jaguar's HTTP address followed by a Dynamo Web site into your browser. For example:

http://jaguar_server_host:8080/mapped_url_name/file_name

Connects your browser to:

- Jaguar’s HTTP port 8080 on the host machine identified by jaguar_server_host.
- The mapped_url_name is the mapping you supply for a Dynamo Web site in the PowerDynamo | Utilities | Configuration | Mappings folder.
• _file_name_ is the file you are accessing from the mapped Dynamo Web site.

### Hot standby

If you have two Jaguar servers, you can enable hot standby which allows you to designate one of the servers as a backup server that accepts client connection requests in case the master server fails. The master server processes client requests. The backup server starts in “Admin” mode and does not accept client requests. If the master server fails or is unreachable, the backup server sets itself to “Ready” mode and accepts client requests. Once the master server is up and accepting requests, the backup server enters “Admin” mode, refusing connections from clients.

**See also** Chapter 4, “Load Balancing, Failover, and Component Availability” for information about component failover.

❖ **To enable hot standby in Jaguar:**

1. You must first enable the two hot standby servers as name servers. Select the Naming Service tab and click Enable as a Name Server to configure the Jaguar server as a name server.

2. Select the Hot Standby tab and click the Enable Hot Standby check box.

3. Enter the Master Server URL using the format iiop://hostname:port. For example, iiop://Jaguar_master:9095.

4. Enter the Backup Server URL using the format iiop://hostname:port. For example, iiop://Jaguar_backup:10000.

   - The master and backup servers must be valid IIOP or IIOPS URLs. You can have only one master and one backup server defined and one of them, but not both, must be defined on the local server.

5. Synchronize the servers using the master as the primary server. See Synchronization in Chapter 3, “Jaguar Clusters and Synchronization”

You can verify the settings of hot standby by checking these properties on the All Properties tab:

- _com.sybase.jaguar.server.CosNaming.nameserver_ must be set to true for both the master and backup servers.

- _com.sybase.jaguar.server.hotstandby_ is set to true if hot standby is enabled.
Chapter 1  Jaguar Configuration

- `com.sybase.jaguar.server.hotstandby.master` is the URL of the hot standby master server.
- `com.sybase.jaguar.server.hotstandby.backup` is the URL of the hot standby backup server.

**Licensing requirements**
Hot standby requires two server deployment licenses, or a separately priced hot-standby license. Please contact your Sybase sales representative for hot-standby licensing and pricing details.

All Properties

For advanced users only. Select this tab to edit server property settings in the Jaguar configuration repository. Properties are listed in the generated HTML documentation for the `Jaguar::Repository` IDL interface, in the description of the `define` method. You can use this tab to edit any property prefixed with “com.sybase.jaguar.server”. Most server properties can be configured on other tabs in the Server Properties dialog box, except the following:

- **com.sybase.jaguar.server.authservice** The name of a custom component that authenticates IIOP user connections. See the *Jaguar CTS Programmer’s Guide* for information on coding a custom authentication service component. The default is empty, which means the server’s built-in authentication features are used.

- **com.sybase.jaguar.server.authorization.service** The name of a custom component that authorizes user access to components and HTTP URLs. See Chapter 35, “Role and Authorization Service Components” in the *Jaguar CTS Programmer’s Guide* for information on coding a custom authorization service component.

- **com.sybase.jaguar.server.authorization.permcachetimeout** The length of time, in seconds, that the server can cache authorization data for a user’s access to a resource. The default is 7200 seconds, which is equivalent to 2 hours.

- **com.sybase.jaguar.server.bindrefresh** Must be set to “start” to allow service components to perform intercomponent calls in their `start` method. The default is “run”, which indicates intercomponent calls cannot be made until the service’s `run` method executes. See Chapter 33, “Creating Service Components” in the *Jaguar CTS Programmer’s Guide* for more information.
• **com.sybase.jaguar.server.http.force.close**  If you are debugging proxy configurations, you can set this property to true to close the HTTP-tunnelled connection after sending a response to each IIOP request. The default is false; setting it to true degrades server performance, and we suggest that you use it only for debugging purposes.

• **com.sybase.jaguar.server.masp.zero-success**  Configures the return codes from MASP method executions. A value of false, the default, specifies that method invocations return 1 for success and 0 for error. A value of true indicates that the meanings of 0 and 1 are reversed. See Appendix A1, “Executing Methods as Stored Procedures” in the *Jaguar CTS Programmer’s Guide* for more information on MASP.

• **com.sybase.jaguar.server.roleservice**  The name of a custom component that evaluates user’s role membership to control access to components and HTTP URLs. See Chapter 35, “Role and Authorization Service Components” in the *Jaguar CTS Programmer’s Guide* for information on coding a custom authorization service component.

• **com.sybase.jaguar.server.services**  A list of components that run as service components in the server. See Chapter 33, “Creating Service Components” in the *Jaguar CTS Programmer’s Guide* for more information.


• **com.sybase.jaguar.server.tx_timeout**  Specifies the default transaction timeout for components running in the server. See Chapter 5, “Defining Components” in the *Jaguar CTS Programmer’s Guide* for more information.

---

**Configuring server stack size**

Your Jaguar server has a stack size property that determines the amount of memory reserved for the call stack associated with each thread created by the server. Jaguar runs each client request on a different thread, so the stack size is the dominant factor in determining how many client requests can be served simultaneously.
Chapter 1    Jaguar Configuration

The default stack size is 256 K. This is appropriate for almost all situations, and provides adequate reserve memory for the worst case loads that have been tested by Sybase engineering and customers.

For production servers that see heavy use from large numbers of clients, you may want to decrease the stack size from the default value. However, you must ensure that the stack size is adequate for the components running on the server. If the stack size is too small, your server may experience thread stack overflow errors, which are recorded in the server log.

Sybase recommends that you do not reduce the stack size if you run components in any of these categories:

• Components that call third-party DLLs or shared libraries
• Java components that call native classes (including JDBC drivers that call out to native libraries)

Warning! Do not reduce the stack size below 64K. If you reduce the stack size, test your server thoroughly under worst-case client loads and check the log for stack overflow errors.

There are different procedures for setting the stack size on UNIX and Windows platforms.

❖ Configuring stack size for servers running on UNIX

1 Highlight the icon for the server and select File | Server Properties.

2 Display the Resources tab. You may need to scroll to the right to see this tab.

3 Enter a stack size in the Thread Stack Size field, specified in bytes as a decimal number. (The field displays with no value if you have not specified a value before. This means the default setting is in effect.)

4 Stop and restart the Jaguar server.

❖ Configuring stack size for servers running on Windows NT

To change the thread stack size, you must have the Microsoft editbin utility, which is included with Microsoft Visual C++. This command-line utility allows you to modify object files, executable files, and dynamic link libraries (DLLs). For more information on the editbin utility, see the Microsoft Visual C++ documentation.

1 Save a copy of the original Jaguar server executable, jagsrv.exe in the Jaguar bin subdirectory.
Managing connection caches

2 Change the thread stack size by running the following command in the Jaguar bin subdirectory:

```
EDITBIN /STACK:value jagsrv.exe
```

where value is the new stack size, specified in bytes. editbin rounds the value up to the closest number divisible by four. For example, the following command sets a 64 KB stack size:

```
EDITBIN /STACK:65536 jagsrv.exe
```

3 You can confirm the new setting by running the Microsoft dumpbin utility, which is included with Microsoft Visual C++. Run the following command:

```
DUMPBIN /HEADERS jagsrv.exe
```

In the output, the stack size appears on a line such as this:

```
100000 size of stack reserve
```

Managing connection caches

A connection cache maintains a pool of available connections that Jaguar components use to interact with third-tier data servers. You must configure connection caches for the specific user and database combinations used by your components. A connection cache entry improves performance by eliminating the overhead associated with setting up a connection when one is required. Create as many connection caches as you need.


A cache must be installed in a server before it is available to components running on that server. You must refresh the cache, refresh the server, or restart the server before any changes to the list of installed caches or to cache properties take effect, and you should test the connection with ping before trying to access it from components.

You cannot define two distinct caches that use identical values for server, user, password, and connectivity library. If two caches are defined with matching values for these settings, and your application requests one, Jaguar returns the first match that is found.
❖ To create a new connection cache:
   1. Highlight the Connection Caches folder.
   2. Select either File | New Connection Cache or File | New XA Connection Cache.
   3. Configure the connection cache properties as described in “Connection cache properties” on page 22.

Configured connection cache entries appear on the right side of the window of Jaguar Manager when you highlight the Connection Cache folder on the left side of the window.

❖ To add a connection cache to a server:
   1. Double-click the Servers icon.
   2. Select the server (listed on the left side of the window) where you want to install a connection cache.
   3. Highlight the Installed Connection Caches folder.
   4. Select File | Install Connection Cache. Then select one of the following options from the Connection Cache wizard:
      • Install an Existing Connection Cache – a list of uninstalled connection caches appears in the dialog box. Highlight the connection cache to be installed and click OK.
      • Create and Install a New Connection Cache – enter the name of the new connection cache to be installed. Select the connection cache property tab and complete the property sheet as described in “Connection cache properties” on page 22.

❖ To view or modify a connection cache entry:
   1. Expand the Connection Caches folder, or, if the cache is installed in a server, expand the Installed Connection Caches folder of that server.
   2. Highlight the connection cache you want to modify.
   3. From the File menu, select one of the following options:
      • Connection Cache Properties – view or modify this connection cache’s properties. See “Connection cache properties” on page 22.
      • Remove Connection Cache – if you have selected a connection cache that is installed on a server, this option removes the connection cache from the server.
Managing connection caches

- Delete Connection Cache – if you have selected a connection cache that is not installed on a server, this option deletes the connection cache from the system.

Connection cache properties

Table 1-7 lists the general and driver connection cache properties that you can configure in Jaguar Manager. After you have configured a connection cache, click OK to save your changes, or click Cancel to disregard them. You must use the cache properties file to manually configure the additional properties described in “Other cache settings” on page 25.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Cache Name</td>
<td>The name for this cache configuration.</td>
<td>Connection cache names are limited to one word, which can contain letters, numbers, and underscores. Names are case-sensitive. You cannot modify the name of an existing connection cache.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the connection cache section.</td>
<td>The description is a string of a maximum of 255 characters.</td>
</tr>
<tr>
<td>Enable Cache-by-Name Access</td>
<td>Select this option to allow retrieval of a database connection using the connection cache name instead of requiring a user name and password.</td>
<td>By default, a cache cannot be retrieved by its name. You must be logged in as jagadmin to update the cache’s properties to allow the cache to be retrieved by name. Cache-by-name is less secure than requiring a user name and password.</td>
</tr>
<tr>
<td>Enable connection sanity check</td>
<td>Whether connections should be verified before releasing them into the cache.</td>
<td>Components may release a connection that is not ready for use by another component. For example, there may be unretrieved results on the connection. Enabling this option causes Jaguar to test whether the connection is usable before replacing it in the cache. Disabling the option increases performance, but may complicate debugging.</td>
</tr>
<tr>
<td>Number of Connections in Cache</td>
<td>The number of connections in the pool.</td>
<td>After a connection is released, it is returned to the pool. The default value is 10. You can increase this number if performance suffers due to an insufficient number of available connections.</td>
</tr>
</tbody>
</table>
When you are defining a Java connection cache that uses the Sybase jConnect driver and that connects to a Sybase DirectConnect™ gateway, set the Service Name field to the back end data server name that the DirectConnect gateway connects to.

Service name is ignored for caches that use JDBC drivers other than jConnect™ for JDBC™.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name</td>
<td>When you are defining a Java connection cache that uses the Sybase jConnect driver and that connects to a Sybase DirectConnect™ gateway, set the Service Name field to the back end data server name that the DirectConnect gateway connects to.</td>
<td>Service name is ignored for caches that use JDBC drivers other than jConnect™ for JDBC™.</td>
</tr>
<tr>
<td>Server Name</td>
<td>ODBC – the ODBC data source name. CTLIB – the server name as it would be specified in a ct_connect call. On UNIX platforms, the server must be listed in the Jaguar interfaces file. For Windows NT, it must be listed in the ini\sql.ini file. JDBC – the URL appropriate for use in JDBC calls. OCI 7.X – the Oracle SQL<em>Net connect string or database alias. OCI 8.X – the Oracle SQL</em>Net connect string or database alias.</td>
<td>For ODBC or JDBC connections, see your driver documentation for more information. For OCI connections, see your Oracle documentation.</td>
</tr>
<tr>
<td>User Name</td>
<td>The user name for this cache.</td>
<td>The name used (along with a password) to connect to the database identified by the server entry.</td>
</tr>
<tr>
<td>Password</td>
<td>The password for this cache.</td>
<td>The password used in connection with a user name to connect to the database identified by the server entry. Passwords are encrypted in the Jaguar configuration file. Jaguar Manager does not display passwords for existing caches. If you need to change a password, enter the new password and click OK.</td>
</tr>
</tbody>
</table>
### Managing connection caches

#### DLL or Class Name

Set the DLL or Class Name and properties using the Driver tab on the Connection Cache Properties window.

Enter the file name of the connection library, and select the connection library type used for this cache.

Your choices for library type are:

- Client Library 11.x – for Sybase Open Client Client-Library connections.
- ODBC – for connections using an open database connectivity driver.
- JDBC – for connections using a Java Database Connectivity driver.
- OCI 7.X – for connections using OCI 7.x
- OCI 8.X – for connections using OCI 8.x

The DLL names for each of the cache types are:

- **NT:**
  - Client Library 11.x – *libjct.dll*. You must use the version in the Jaguar dll subdirectory. This version has been optimized for Jaguar threading.
  - ODBC – *odbc32.dll*
  - OCI 7.X – *ociw32.dll*
  - OCI 8.X – *oci.dll*

JDBC – the Java class name for the driver class. For example, the Sybase jConnect 5.2 driver requires *com.sybase.jdbc2.jdbc.SybConnectionPoolDataSource*.

The file must be in your PATH environment variable for ODBC, Client-Library, or OCI caches, and in CLASSPATH for JDBC caches.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLL or Class Name</td>
<td>Set the DLL or Class Name and properties using the Driver tab on the Connection Cache Properties window.</td>
<td>The DLL names for each of the cache types are:</td>
</tr>
<tr>
<td></td>
<td>Enter the file name of the connection library, and select the connection library type used for this cache.</td>
<td>• NT:</td>
</tr>
<tr>
<td></td>
<td>Your choices for library type are:</td>
<td>- Client Library 11.x – <em>libjct.dll</em>. You must use the version in the Jaguar dll subdirectory. This version has been optimized for Jaguar threading.</td>
</tr>
<tr>
<td></td>
<td>• Client Library 11.x – for Sybase Open Client Client-Library connections.</td>
<td>- ODBC – <em>odbc32.dll</em></td>
</tr>
<tr>
<td></td>
<td>• ODBC – for connections using an open database connectivity driver.</td>
<td>- OCI 7.X – <em>ociw32.dll</em></td>
</tr>
<tr>
<td></td>
<td>• JDBC – for connections using a Java Database Connectivity driver.</td>
<td>- OCI 8.X – <em>oci.dll</em></td>
</tr>
<tr>
<td></td>
<td>• OCI 7.X – for connections using OCI 7.x</td>
<td>JDBC – the Java class name for the driver class. For example, the Sybase jConnect 5.2 driver requires <em>com.sybase.jdbc2.jdbc.SybConnectionPoolDataSource</em>.</td>
</tr>
<tr>
<td></td>
<td>• OCI 8.X – for connections using OCI 8.x</td>
<td>The file must be in your PATH environment variable for ODBC, Client-Library, or OCI caches, and in CLASSPATH for JDBC caches.</td>
</tr>
</tbody>
</table>
### Other cache settings

You cannot use Jaguar Manager to set the cache settings described in this section. You must edit the underlying configuration file to change them. Use a text editor to edit the cache's property file in `Repository/ConnCache`. The file is `CacheName.props`, where `CacheName` represents the cache name as it displays in Jaguar Manager.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLL or Class Name</td>
<td>• For UNIX platforms:&lt;br&gt;Client Library 11.x – <code>libjct_r.so</code> for Solaris, AIX, Digital UNIX, and LINUX; <code>libjct_r.sl</code> for HP-UX. You must use the version in the Jaguar <code>lib</code> subdirectory. This version has been optimized for Jaguar threading.&lt;br&gt;ODBC – <code>libodbc.so</code> (installed in <code>$JAGUAR/intersolv/odbc/dlls</code>) for Solaris and AIX; <code>libodbc.so</code> (installed in <code>/var/opt/DAU100/connect/lib</code>) for Digital UNIX; <code>libodbc.sl</code> (installed in <code>$JAGUAR/intersolv/odbc/dlls</code>) for HP-UX.&lt;br&gt;OCI 7.X or OCI 8.X – <code>libclntsh.so</code> for Solaris, AIX, Digital UNIX, and LINUX; <code>libclntsh.sl</code> for HP-UX.&lt;br&gt;Client Library 11.x – <code>libjct.dll</code>. You must use the version in the Jaguar <code>dll</code> subdirectory. This version has been optimized for Jaguar threading.&lt;br&gt;ODBC – <code>odbc32.dll</code>&lt;br&gt;OCI 7.X – <code>ociw32.dll</code>&lt;br&gt;OCI 8.X – <code>oci.dll</code>&lt;br&gt;JDBC – the Java class name for the driver class. For example, the JDBC-ODBC driver requires <code>sun.jdbc.odbc.JdbcOdbcDriver</code>. On Solaris, Digital UNIX, and LINUX, the file must be in your <code>LD_LIBRARY_PATH</code> environment variable for ODBC, Client-Library, or OCI caches, and in <code>CLASSPATH</code> for JDBC caches.&lt;br&gt;On AIX, the file must be in your <code>LIBPATH</code> environment variable for ODBC, Client-Library, or OCI caches, and in <code>CLASSPATH</code> for JDBC caches.&lt;br&gt;On HP-UX, the file must be in your <code>SHLIB_PATH</code> environment variable for ODBC, Client-Library, or OCI caches, and in <code>CLASSPATH</code> for JDBC caches.</td>
</tr>
</tbody>
</table>
Managing connection caches

JDBC connection properties

Different JDBC drivers recognize different sets of properties. For a JDBC connection cache, these additional connection properties allow you to specify settings beyond those shown in the Connection Cache Properties dialog.

Cache properties are defined in the form:

    propname=value specification

Any property whose name does not begin with `com.sybase.jaguar` is passed to the JDBC driver as a connection property. For example:

    PACKETSIZE=2048

If a property setting conflicts with a setting in the Connection Cache Properties dialog, the dialog setting takes precedence.

See your driver documentation for more information.

Enabling set-proxy support

Current versions of Adaptive Server Enterprise and Adaptive Server Anywhere allow a user to assume the identity and privileges of another user. Except for Oracle connections that use OCI, you can use this feature with any database that recognizes this command:

    set session authorization "login-name"

When proxy support is enabled, connections retrieved from the cache are set to act as a proxy for the user name associated with the Jaguar client. To set proxy to another user name, use the Java JCMCache.getProxyConnection() method or the C JagCmGetProxyConnection() routine in your component.

You must enable set-proxy support in the cache properties file before components can take advantage of it. To enable set-proxy support, add the following line to the cache properties file:

    com.sybase.jaguar.conncache.ssa=true

To disable support, delete this line or change `true` to `false`.

Client-Library connection settings

You can set the following properties for Client-Library connections in the `$JAGUAR/Repository/ConnCache/<cachename>.props` file. See the Client-Library documentation for the ct_con_props routine for more information:

- CS_HOSTNAME
- CS_APPNAME
• CS_TDS_VERSION
• CS_PACKETSIZE

This is an example of a properties file that sets all of these properties:

```
# this is a comment line
CS_HOSTNAME=myhostname
CS_TDS_VERSION=CS_TDS_46
CS_PACKETSIZE=512
CS_APPNAME=myapp
```

Any property name not recognized by Jaguar is ignored.

The CS_TDS_VERSION property takes one of these values; CS_TDS_50, CS_TDS_495, CS_TDS_46, CS_TDS_42, or CS_TDS_40.

The CS_PACKETSIZE property takes a numeric value within the range of legal packet sizes for the server. If the server does not support the packet size then the cache will not be able to connect to the server.

**Note** Make sure there is no extraneous white space between the property name, the equals sign, and the property value, or after the property value.

**Connection cache refresh**

If you have just installed the cache in a server or modified an installed cache, refresh the server or the connection cache before you attempt to test the cache. You can refresh as follows:

• To refresh the server, highlight the server icon where the cache is installed, then choose File | Refresh. All caches installed in the server will be refreshed.

• To refresh the cache:
  a Highlight the Installed Connection Caches folder under the server icon where the cache is installed.
  b Highlight the cache’s icon and choose File | Connection Cache Properties.
  c In the Connection Cache Properties dialog, click Refresh.

Refreshing a cache may affect running components that are using the cache, specifically:
Managing connection caches

- If you change the connectivity library setting, cache references held by components become invalid. Attempts to retrieve connections or query cache properties will cause errors. In this case, the component must retrieve a new cache handle.

- If you change other properties, such as user name, password, server name, or the number of connections in a cache, cache references remain valid, but components may be affected by the changed settings. For example, if you change the server name, connections retrieved after the cache has been refreshed will go to the server indicated by the new name.

Connection cache ping

This feature allows you to test the cache configuration to verify that connections can be made using the supplied parameters. To ping, the connection must be installed in the server that Jaguar Manager is connected to. If you have just installed the cache or changed any settings, refresh the cache before testing it.

To test the cache with ping:

1. Highlight the Installed Connection Caches folder under the server icon where the cache is installed.

2. Highlight the cache’s icon and choose File | Connection Cache Properties.

3. In the Connection Cache Properties dialog, enter the password. (Though the password is recorded in the repository, the window does not display it.)

4. Click Ping.

5. Jaguar Manager reports whether the connection attempt succeeded.

If ping fails, check the message text for a description of the problem. The server log file may contain additional information about the cause of the error.

If you change the cache properties to correct the problem, you must refresh the cache before testing again.
Managing identities

Identities define user names and passwords used for intercomponent calls. If a component’s Run-As Identity property is set to Specified, then intercomponent calls issued by the component use the user name and password defined in a Jaguar Manager identity.

❖ To create an identity:
   1 Highlight the Identities folder in Jaguar Manager.
   2 Choose File | New Identity.
   3 Enter a name for the new identity and click Create New Identity, then enter the identity properties described in “Identity properties” on page 29.

❖ To view or change an identity:
   1 Expand the Identities folder.
   2 Highlight the identity of interest.
   3 Choose File | Identity Properties.
   4 Edit identity properties. See “Identity properties” on page 29 for more information.

❖ To delete an identity:
   1 Expand the Identities folder.
   2 Highlight the identity of interest.
   3 Choose File | Delete Identity.

Identity properties

The Identity Properties dialog has three tabs: Basic, SSL, and Entrust.

The Basic tab is where you specify the user name and password for the identity.

Use the SSL tab to define whether SSL is used for connections made using the identity, and if so, the characteristics of the SSL session.
Managing XA resources

• **Security characteristic**  If set to none, SSL is not used. Otherwise, specifies the minimum level of SSL security that connections must support. The available settings depend on whether your installation supports full-strength U.S. domestic encryption, or the export-legal level of encryption. See Chapter 5, “Security Configuration” for more information.

• **Description**  Jaguar Manager fills in this field with a description of the specified security characteristic.

• **Certificate Label**  For connections that require mutual authentication, specifies the name of a client certificate to use.

• **Use Entrust**  Whether to use an Entrust user profile as a client certificate for connections that require mutual authentication. Jaguar does not come with Entrust software, but if Entrust is installed at your site, you can configure the identity to use an Entrust ID. If this check box is selected the Certificate Label setting is ignored.

The settings on the Entrust tab are ignored unless you have also selected the Use Entrust check box on the SSL tab.

• **Entrust INI File**  The full path to the `entrust.ini` file.

• **Entrust User Profile**  The full path to an Entrust user profile (.epf file).

• **Entrust Password**  The password that opens the specified Entrust user profile.

Managing XA resources

You must configure XA resources to access a specific database. XA resources differ from connection caches in that XA resources are XA-Library interfaces that maintain their own connection pool separate from the connection cache connection pool.

If a get connection call (such as the Java `getConnection` or the C `JagCmGetConnection` method call) is in a transaction, the XA resource is automatically used to return a connection. If a get connection call is not in a transaction, the connection cache is automatically used. If the transactional behavior for a component uses the Supported option, then Jaguar determines at runtime whether the component executes its get connection calls in a transaction; if it does, you must configure both a connection cache and a corresponding XA resource for a database.
Chapter 1    Jaguar Configuration

By default, Jaguar uses these shared libraries or DLLs to obtain an XA resource that is exported from the database connection libraries:

<table>
<thead>
<tr>
<th>Connection library</th>
<th>Shared library</th>
<th>DLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybase Client Library 11.0</td>
<td>libjxa.so</td>
<td>libjxa.dll</td>
</tr>
<tr>
<td>Oracle OCI 7.x</td>
<td>libclntsh.so</td>
<td>xa73.dll</td>
</tr>
<tr>
<td>Oracle OCI 8.x</td>
<td>libclntsh.so</td>
<td>xa80.dll</td>
</tr>
</tbody>
</table>

To use a shared library or DLL other than the default, edit the connection cache properties file `%JAGUAR%\Repository\ConnCache\<cache_name>.props`. For example, to instruct Jaguar to use `oraclient8.dll` instead of `xa80.dll` for Oracle OCI 8.1.x, add this line to the connection cache properties file:

```
com.sybase.jaguar.conncache.xadllname = oraclient8.dll
```

**Note** You must install XA resources in a Jaguar server before components in that server can access the XA resources. You must refresh XA resources, refresh the server, or restart the server before any changes to the list of installed XA resources or to XA resource properties take effect.

If you execute a transaction without an XA resource configured for a database, the Jaguar connection manager returns `CS_FAIL`.

If a configured XA resource is not running or cannot be connected to, the Jaguar server cannot initialize. To solve this problem, perform one of these tasks:

- Start the XA resource.
- Make the XA resource available to the Jaguar server.
- Start the JagRepair server and delete the XA resource. See *Getting Started* for your platform for more information about JagRepair.

❖ **To create OTS/XA transactions for XA resources:**

1. Right-click the Installed Connection Caches folder.
2. Select Install XA Resource.
3. Select either Install an Existing XA Resource, or Create and Install a New XA Resource.
4. In the XA resource properties dialog, enter:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Description</td>
<td>A brief phrase describing the purpose of the XA resource.</td>
</tr>
</tbody>
</table>
Managing XA resources

<table>
<thead>
<tr>
<th>Tab</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Server Name           | Name of the XA resource server. For JDBC, enter: `NetworkProtocol=Tds:Server=yourServer:Port=TdsPort#` where:  
  `yourServer` is the server that the driver is installed on.  
  `TdsPort#` is the port number where TDS calls are received. |
| User Name             | A name you can use to access the server. |
| Password              | The password for the user. |
| DLL or Class Name     | The file name of the XA resource library. For JDBC, enter: `com.sybase.jdbc2.jdbc.SybXADataSource` |
| Client Library 11.0   | Connections made via Sybase Open Client Client-Library version 11.1 (included with Jaguar). Client-Library is the proprietary, native C/C++ interface to Sybase databases and database gateways. |
| OCI 7.x               | Oracle Call Interface version 7.x. OCI is the proprietary, native driver for C/C++ access to Oracle databases. |
| OCI 8.x               | OCI version 8.x or later. |
| JDBC                  | A Java Database Connectivity driver such as Sybase jConnect. |
| Database Name         | If you selected CT-LIB, OCI 7.x, or 8.x, specify the database name. |
| Default Open String   | The string used to connect to the XA resource. You cannot modify this string, which is automatically built from the information that you entered in the previous tabs. |
| Open String Suffix    | In this optional field, you can specify any valid open string options. For example, for a Sybase Client-Library 11.0 XA resource, you can enter: `-L logfile` where `logfile` is where you want to store log information. |
| Close String          | In this optional field, you can specify a value used by the resource to close a connection. |

See your XA resource documentation for more information about the Open String Suffix and Close String syntax.

5 Click Refresh.

**Note** If the Open String is set incorrectly, the Jaguar server does not initialize.
Using File Viewer and the Runtime Monitor

The File Viewer and the Runtime Monitor allow you to track a Jaguar server’s performance and statistics.

File Viewer

The File Viewer allows you to monitor:

- **REQUESTLOG** – the `httprequest.log` file described in “HTTP logging and statistics” on page 5.
- **SRVLOG** – the `srv.log` file tracks server events and any configured tracing events. See “Log/Trace” on page 8 for information regarding configuration of the log file and trace flags.
- **ERRORLOG** – the `httperror.log` file tracks HTTP errors, such as a request for an HTML file that does not exist. If you define additional servers, the name of the log file is prepended with the server name. For example, if you create a server named Test_server, then HTTP error messages for that server are directed to the `Test_serverhttperror.log` file.

Before you bring up File Viewer, make sure that a listener is configured with a protocol of TDS and a port number of 7878. See “Preconfigured listeners” on page 129 for more information.

❖ To bring up File Viewer:

1. Double-click the server for which you want to view files.
2. Highlight the File Viewer icon.

When the File Viewer appears, configure the following items:

- **File** – select the file you want to view from the File drop-down list.
- **Start/Stop** – the Start/Stop button allows you to start and stop real-time viewing.
- **Refresh** – select the frequency of the refresh rate from the drop-down list.
- **Previous/Next** – allows you to scroll through the current file if it is too large to view or if you have selected a start position that does not start at the beginning of the file.
- **File Size** – this field displays the current file size in bytes.
Using File Viewer and the Runtime Monitor

- Start Position – select the start position of the file you are viewing. Your options are:
  - Tail – display as much of the end of the file as fits on the screen.
  - End – clear the file display. When you click Start, only the new entries into the file will display.
  - Top – display the file starting from the top.
  - Specify Position – allows you to select a starting position by positioning a slider. Choose the incremental position from the beginning of the file.

Runtime Monitor

The Runtime Monitor allows you to monitor server events and statistics, which may help you anticipate and prevent server problems.

❖ To start the Runtime Monitor:
  1. Double-click the Servers icon.
  2. Double-click the server you want to monitor.

  The Runtime Monitor can connect to other Jaguar servers via an IIOP listener. The server configuration identifies the host and port number to which the Runtime Monitor attempts to connect.

  3. Click the Runtime Monitor icon.

The Runtime Monitor displays the folders listed below. For each of the top-level groups, there are subgroup folders. Click the subgroup folder whose statistics you want to view.

- Packages – monitor events and statistics for a specific package, or for all packages on the server.
- Connection caches – monitor a specific connection cache or statistics for all caches.
- Network – monitor protocol-specific session information.

When viewing a subgroup’s counters, you can change the view by selecting any of these options from the File menu:

- Refresh – refresh the display to obtain the latest counters.
- Per-Second – display the per-second values.
• Counter – display the accumulated value.
• View Values... – view the values in a separate dialog that automatically refreshes according to the refresh rate, which you can select from a drop-down list. If the group contains subgroups, expand those subgroups until you see the values.

If the subgroup contains counters, the current values of the counters display on the right side of the window.

There are two types of counters:
• Snapshot counters – represent values at a particular point in time that are likely to either increase or decrease. For example, number of sessions, number of instances active, number of active connections, and so on. Snapshot counters do not display information when set at the per-second rate; instead they display “N/A.”
• Cumulative counters – represent values that always increment and never decrement. For example, number of invocations, number of connections opened, number of network requests, bytes read and written, and so on. Cumulative counters display both per-second and counter information.

Using the OTS Transaction Monitor

You can monitor and administer transactions on a single Jaguar server from Jaguar Manager. You must be connected to the Jaguar server before you can monitor or administer it.

❖ To administer OTS/XA transactions:

1 Select OTS Transaction Viewer for a server. The dialog displays the following information for all incomplete transactions. Because the OTS Transaction Viewer captures the information about incomplete transactions at a particular moment, you need to refresh the OTS Transaction Viewer to see any changes in the incomplete transaction status or any changes that you have made directly in the OTS Transaction Viewer.

• Transaction ID  The local identifier associated with the transaction on a specific server.
Using the OTS Transaction Monitor

- **Family ID**  The identifier of the root transaction that this transaction belongs to.

  **Note** Because Jaguar does not support nested transactions, the Family ID is the same as the Transaction ID.

- **State**  Describes the status of the transaction and can be:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>local_state_active</td>
<td>The transaction is currently active in the server.</td>
</tr>
<tr>
<td>local_state_inactive</td>
<td>The transaction is not currently active in the server.</td>
</tr>
<tr>
<td>local_state_preparing</td>
<td>The transaction is in the process of being prepared.</td>
</tr>
<tr>
<td>local_state_prepared</td>
<td>The transaction has been prepared.</td>
</tr>
<tr>
<td>local_state_committing</td>
<td>The transaction is in the process of being committed.</td>
</tr>
<tr>
<td>local_state_committed</td>
<td>The transaction has been committed.</td>
</tr>
<tr>
<td>local_state_commit_complete</td>
<td>The transaction has committed and all participants have been informed, but the outcome may not have been reported to the transaction originator. For example, there may have been heuristic outcomes.</td>
</tr>
<tr>
<td>local_state_before_abort</td>
<td>The transaction has rolled back but has not yet started the rollback process.</td>
</tr>
<tr>
<td>local_state_aborting</td>
<td>The transaction is in the process of rolling back.</td>
</tr>
<tr>
<td>local_state_aborted</td>
<td>The transaction has been rolled back.</td>
</tr>
<tr>
<td>local_state_abort_complete</td>
<td>The transaction has rolled back and all participants have been informed, but the outcome may not have been reported to the transaction originator. For example, there may have been heuristic outcomes.</td>
</tr>
<tr>
<td>local_state_finished</td>
<td>The transaction has completed.</td>
</tr>
<tr>
<td>local_state_none</td>
<td>The server knows about the transaction, but the server is not a participant in the transaction.</td>
</tr>
<tr>
<td>local_state_present</td>
<td>The transaction is active in the server but is not yet a participant in the transaction.</td>
</tr>
</tbody>
</table>

  **Note** You can execute Rollback only on transactions that are not yet in the prepared state. You can execute all other operations only on transactions that have been prepared.

- **Lock Holder**  The transaction that holds the lock.

- **Lock Waiter**  The transaction that is waiting for a lock to be released.
• **Level** An integer that represents the level of a nested transaction. Since Jaguar does not support nested transactions, this field is always 1.

2 In the dialog that displays, select a transaction and click one of these buttons:

- **Refresh List** Refresh the list of all unfinished transactions.
- **Rollback** Roll back the entire transaction, including any sub-transactions.

**Note** Jaguar does not currently support nested OTS/XA transactions (also called subtransactions).

- **Force Complete** Force the transaction’s rollback or commit processing (whichever outcome the transaction coordinator determined for the transaction) to be completed.
- **Heuristic Commit** Perform a heuristic commit.
- **Heuristic Rollback** Perform a heuristic rollback.
- **Global Info** Displays the global transaction identifier and the identifier of the application that started the transaction.

---

### Setting up the message service

The message service allows you to publish or send messages to a queue, where they are stored until they can be delivered to the message queue’s recipient, which is either a client or a component. You can also subscribe to scheduled messages.

The message service provides transient and persistent message storage for message consumers and allows message producers to send messages to a particular message queue, or to publish messages with specific topics, available to all message queues.

For persistent message storage, use a database that is tuned for frequent inserts and deletes. For example, if you are using Adaptive Server Enterprise, use a database other than the master database and ensure that the transaction log is large enough to accommodate frequent transactions. See your database documentation for more information.
After you have configured the message service, it starts automatically when you start a Jaguar server.


❖ **To configure the message service:**

1. Make these changes to the *MessageService.props* file, located in
   - %JAGUAR%\Repository\Component\CtsComponents (NT)
   - $JAGUAR/Repository/Component/CtsComponents (UNIX):
     - To specify the connection cache, change the line
       `cms.cache=MessageServiceCache` to `cms.cache=yourCache`, where
       `yourCache` is the name of a user-defined JDBC connection cache that
       allows cache-by-name access.
     - To enable the JCM_FORCE option for access to the connection
       cache, change the line `cms.force=false` to `cms.force=true`.
     - To enable message service debugging at server start-up, change the
       line `cms.debug=false` to `cms.debug=true`.
     - If your database server does not accept the default syntax, change the
       SQL commands in the *XX.YYYYYY* statements. If the database does
       not have an image type, change image to unbounded (large) binary. If
       the database driver cannot handle character-set conversions, change
       the datatype of the varchar columns in the SQL schemas to varbinary.

2. On the All Properties tab in Jaguar Manager’s Server Properties dialog, add “CtsComponents/MessageService” to the
   `com.sybase.jaguar.server.services` server property.

3. If your server is in a cluster:
   - For each server in the cluster, copy *MessageService.props* to
     `$JAGUAR/Repository/Component/CtsComponents`.
   - **Synchronize the cluster.** See Chapter 3, “Jaguar Clusters and
     Synchronization”.

4. Restart the Jaguar server. Message service starts automatically when you
   start the server.
Administering the message service

You can administer the message service using the graphical interface.

❖ To start the message service graphical interface:

1 Run one of these scripts:
   • UNIX – $JAGUAR/html/classes/msgadmin.sh
   • NT – %JAGUAR%html\classes\msgadmin.bat

2 In the login window, enter the:
   • User Name Name authorized to access the message service.
   • Password Password for the user name.
   • Host Name Name of the machine where the message service resides.
   • Port Number Port number through which the connection to the message service is made.
   • Service CtsComponents\MessageService

3 Click Connect.

To enable debugging, in the Message Service Administration window, click Enable Debugging.

Adding the message service parts

Once you have configured the message service, add the message service parts using the Message Service Administration window:

• Message queues – add one message queue for each message recipient.
• Message topics – to identify the subject of messages that you publish or send, define message topics.
• Message queue selectors – to specify the topics of the published messages you want to receive, add a message queue selector for each topic.
• Listeners – to notify a component when a message arrives in its queue, add a listener for each message queue.
• Access roles – to restrict access to a message queue or to grant one user access to another user’s queue, add access roles for the message queue. To restrict access to messages with a particular topic, add access roles for the topic.
• Thread pools – define thread pools to handle client and component notification.

❖ To add a message queue:
  1 Select the Configured Queues tab.
  2 Click Add.
  3 In the New Queue dialog, enter a name for the message queue.
  4 Click OK.

❖ To add a topic:
  1 Select the Configured Topics tab.
  2 Click Add.
  3 In the New Topic dialog, enter the topic name.
  4 Click OK.

❖ To add a message queue selector:
  1 If you are adding a selector to an active queue, select the Active Queues tab; if you are adding a selector to an inactive queue, select the Configured Queues tab.
  2 Select a message queue from the list.
  3 Click Properties.
  4 In the Properties dialog, click on the Selectors tab.
  5 Click Add.
  6 In the Add Selector dialog, enter the message queue selector. A selector is an expression that contains an equality condition for a topic and conforms to the Java Message Service (JMS) selector specification, which is a subset of the SQL-92 syntax. For example, to receive all published messages with the topic “StockPrice.SYBS”, add this selector to the message queue:

    "topic = 'StockPrice.SYBS'"

You can also set the selector to “TRUE” to match all message topics.
  7 Click OK.
❖ To add a listener for a message queue:
1. If you are adding a listener for an active queue, select the Active Queues tab; if you are adding a listener for an inactive queue, select the Configured Queues tab.
2. Select a message queue from the list.
3. Click Properties.
4. In the Properties dialog, click on the Listeners tab.
5. Click Add.
6. In the Add Listener dialog, enter a listener that has been installed in Jaguar, in this format: package_name/component_name.
7. Click OK.

❖ To add an access role for a message queue or message topic:
1. To add an access role for an active queue, select the Active Queues tab; to add an access role for an inactive queue, select the Configured Queues tab. To add an access role for an active topic, select the Active Topics tab; to add an access role for an inactive topic, select the Configured Topics tab.
2. Select a queue or topic from the list.
3. Click Properties.
4. In the Properties dialog, select the Roles tab.
5. Click Add.
6. In the Add Role dialog, enter the role name. You can use a wildcard character ("*") when the queue or topic does not require a specific role. For subtopics that do not require a specific role, the name can end with the wildcard character.
7. Select the role type:
   • Consumer To receive messages from the queue, or to receive messages with the specified topic.
   • Producer To publish or send messages to the queue, or to publish or send messages with the specified topic.
   • Security To administer the message queue or topic.
8. Click OK.
If more than one access role is assigned to a message queue or message topic, a client needs to possess only one of the roles to access the queue or topic. If a message queue or topic has no assigned roles, any client can access it.

❖ To add a thread pool:
1. Select the Thread Pool tab.
2. Click Add.
3. In the New Thread Pool dialog, enter the name of the thread pool.
4. Click OK.

Configuring message queue and thread pool properties

❖ To edit a message queue’s configuration properties:
1. To edit properties for an active queue, select the Active Queues tab; to edit properties for an inactive queue, select the Configured Queues tab.
2. Select a queue from the list.
3. Click Properties.
4. In the Properties dialog, select the Configuration tab.
5. Edit the configuration properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Datatype</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum</td>
<td>long</td>
<td>0</td>
<td>The maximum number of messages held in the queue; if set to zero or a negative number, there is no limit.</td>
</tr>
<tr>
<td>timeout</td>
<td>long</td>
<td>0</td>
<td>The number of seconds the queue remains in memory when it is not being accessed by either a client or a listener; set to zero or a negative number for no timeout.</td>
</tr>
<tr>
<td>receive</td>
<td>boolean</td>
<td>true</td>
<td>Indicates whether clients can receive messages from the queue. This is useful for shared queues that are accessed with list and move operations.</td>
</tr>
<tr>
<td>store</td>
<td>boolean</td>
<td>true</td>
<td>Indicates whether to retain persistent messages when the queue is inactive.</td>
</tr>
<tr>
<td>qop</td>
<td>string</td>
<td>“none”</td>
<td>Indicates the quality of protection required for the message queue object.</td>
</tr>
</tbody>
</table>

6. Click Set.
❖ To deactivate a message queue:
1 Select the Active Queues tab.
2 Select a queue from the list.
3 Click Close.

❖ To modify the number of threads in a thread pool:
1 Select the Thread Pool tab.
2 Select a thread pool from the list.
3 Click Properties.
4 Edit the number of threads.
5 Click Save.

Property | Datatype | Default value | Description
--- | --- | --- | ---
readers | long | 0 | The number of reader threads in the thread pool, which are used for client notification.
writers | long | 0 | The number of writer threads in the thread pool, which are used for client notification.
worker | long | 0 | The number of worker threads in the thread pool, which are used for component notification.

Viewing messages and statistics for queues and topics

❖ To view a list of the messages in a queue:
1 To view messages in an active queue, select the Active Queues tab; to view messages in an inactive queue, select the Configured Queues tab.
2 Select a message queue from the list.
3 Click List.

❖ To view statistics for all message queues and topics:
• Select the Statistics tab.

❖ To view a message queue’s statistics:
1 To view statistics for an active queue, select the Active Queues tab; to view statistics for an inactive queue, select the Configured Queues tab.
2 Select a message queue from the list.
Setting up the message service

3 Click Properties.
4 In the Properties dialog, select the Statistics tab.

❖ To view a message topic’s statistics:
1 To view the statistics of an active topic, select the Active Topics tab; to view the statistics of an inactive topic, select the Configured Topics tab.
2 Select a topic from the list.
3 Click Properties.
4 In the Properties dialog, select the Statistics tab.

Deleting message service parts

❖ To delete all the messages in a queue:
1 Select the Active Queues tab.
2 Select a queue from the list.
3 Click Flush.

❖ To delete a message queue:
1 If you are deleting an active queue, select the Active Queues tab; if you are deleting an inactive queue, select the Configured Queues tab.
2 Select a queue from the list.
3 Click Delete.
4 In the Delete Queue confirmation dialog, click Yes.

If you are deleting an active queue, it must also be configured, otherwise, there is nothing to delete.

❖ To delete a topic:
1 Select the Configured Topics tab.
2 Select a topic from the list.
3 Click Delete.
4 In the Delete Topic confirmation dialog, click Yes.
❖ To delete an access role from a message queue:
1 If you are deleting an access role from an active queue, select the Active Queues tab; if you are deleting an access role from an inactive queue, select the Configured Queues tab.
2 Select a message queue from the list.
3 Click Properties.
4 In the Properties dialog, select the Role tab.
5 Select a role from the list.
6 Click Delete.
7 In the Delete Role confirmation dialog, click Yes.

❖ To delete a message queue selector:
1 If you are deleting a selector from an active queue, select the Active Queues tab; if you are deleting a selector from an inactive queue, select the Configured Queues tab.
2 Select a message queue from the list.
3 Click Properties.
4 In the Properties dialog, click on the Selectors tab.
5 Choose a selector from the list.
6 Click Delete.
7 In the Delete Selector confirmation dialog, click Yes.

❖ To delete a thread pool:
1 Select the Thread Pools tab.
2 Select a thread pool from the list.
3 Click Delete.
4 In the Delete Thread Pool confirmation dialog, click Yes.

❖ To delete a listener:
1 If you are deleting a listener from an active queue, select the Active Queues tab; if you are deleting a listener from an inactive queue, select the Configured Queues tab.
2 Select a message queue from the list.
Setting up the message service

3 Click Properties.
4 In the Properties dialog, click on the Listeners tab.
5 Choose a listener from the list.
6 Click Delete.
7 In the Delete Listener confirmation dialog, click Yes.

Closing the graphical interface

To close the message service graphical interface, in the Message Service Administration window, click Exit.
CHAPTER 2

Jaguar Naming Services

A naming service lets you associate a logical name with an object, such as a package and component. Naming helps Jaguar applications easily locate an object anywhere on a network, then implement the referenced object.

The naming service “binds” a name to an object. The combination of bound name and its referenced object is the name context. The referenced object in a name context can be a component within a package or even an existing name context, the same way a named directory can contain a file or other named directory.

The collection of name context information—each object and its bound name—comprises the namespace. When client applications reference an object, they look to the namespace to cross-reference or resolve the name with the referenced object.

### How does the Jaguar naming service work?

The process of binding objects is performed by a name server. Each Jaguar server can be its own name server, or you can configure a Jaguar server to use another server as its name server. You can also use an external naming service, such as an LDAP server, in conjunction with Jaguar’s naming service.

You set the naming service options for each Jaguar server using the Naming Service tab on the Server Properties window.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How does the Jaguar naming service work?</td>
<td>47</td>
</tr>
<tr>
<td>CORBA CosNaming API support</td>
<td>51</td>
</tr>
<tr>
<td>JNDI support</td>
<td>55</td>
</tr>
<tr>
<td>Configuring the Jaguar naming service</td>
<td>60</td>
</tr>
<tr>
<td>Using an LDAP server with Jaguar</td>
<td>61</td>
</tr>
</tbody>
</table>
How does the Jaguar naming service work?

Jaguar server initial context

The Jaguar naming service relies on an “initial” or default name context for each Jaguar server. You set the initial context when you set up the Jaguar Naming Service properties.

The server name context syntax follows a specific organization or schema. You can use this schema to represent the hierarchy of objects in the namespace, for example by geographic region, organizational unit, and so on.

If you use a Jaguar server as the name server, and do not use an external naming service, the initial context for your Jaguar server uses this format:

\(<\text{Level 1}>/\text{Level 2}>/\text{Level 3}>/\ldots\)

The number of levels depends on the hierarchy you want to represent. For example:

- US/sybase/finance
- US/sybase/marketing
- US/sybase/sales

If you use an LDAP server as an external naming service, the initial context must follow the syntax and schema of the LDAP server. LDAP servers have predefined schema for common objects such as country, organization, and organizational unit. Jaguar uses the following format for an LDAP-compatible initial context:

\(ou=<\text{organizational unit}>, o=<\text{organization}>, c=<\text{country}>\)

Using the previous examples, the initial contexts would be:

- ou=finance, o=sybase, c=US
- ou=marketing, o=sybase, c=US
- ou=sales, o=sybase, c=US

On start-up, the name server binds all object implementations on a Jaguar server to the initial context of the server on which the object is installed. Once the server binds an object, the structure of the resulting name context is:

\(<\text{initial context}>/\langle\text{package}\rangle/\langle\text{component}\rangle\)

where

\(<\text{initial context}>\) is the initial context property for the server where the component is installed.

\(<\text{package}>\) is the name of the package being bound, as displayed in Jaguar Manager.
Chapter 2   Jaguar Naming Services

<component> is the name of the component being bound, as displayed in Jaguar Manager.

If you have multiple Jaguar servers that use one Jaguar server as their name server, the name server binds the objects on those servers using the initial contexts of their respective servers. If you do not specify an initial context for any of those servers, the name server binds the objects with the initial context of the designated name server.

**Note** You can set the server properties to enable password protection for name binding on a Jaguar name server. See “Name binding password security” on page 61.

**Name binding example**

To illustrate how a Jaguar name server uses the initial context to create name contexts for objects on multiple servers, let’s say you have three Jaguar servers:

- Server A contains package Pkg1 and components CompX and CompY. You assign the server an initial context of /us/sybase/serverA.
- Server B contains package Pkg2 and the component CompZ. You assign the server an initial context of /us/sybase/serverB.
- Designate server C to be the name server for servers A and B by specifying the URL for server C (iiop://myhost:9050) in their Naming Services properties.

When you start server A, it connects to server C, using the name server URL you entered in server A’s Naming Service properties. The name server gets the initial context for server A and binds each object installed on server A. The resulting name contexts are based on server A’s initial context, the package name, and the components in the package. For this example, the name server creates the following bindings:

/us/sybase/serverA/Pkg1/CompX
/us/sybase/serverA/Pkg1/CompY

If you are using an external naming service such as LDAP, the name server also updates the existing object references on that server, if any.

When you start server B, the name server creates the following binding:

/us/sybase/serverB/Pkg2/CompZ.
How does the Jaguar naming service work?

Figure 2-1 illustrates the name binding process.

*Figure 2-1: Name binding process*

An application referencing object CompY uses the URL of the name server, followed by the object’s name context. For example:

```
iiop://myhost:9050/us/sybase/serverA/Pkg1/CompY
```

The name server finds the name context in the namespace, resolves the name context with the object it references, then implements the object.

If you had not assigned an initial context to Server A, the name server, server C, would create name contexts for objects Pkg1/CompX and Pkg1/CompY using the initial context of the name server. In this case, the client application can simply retrieve CompY using this URL:

```
iiop://myhost:9050/Pkg1/CompY
```
Chapter 2  Jaguar Naming Services

Transient vs. persistent storage

Jaguar’s naming service inherently provides transient object name storage. The name server is instantiated when you start a Jaguar server, and binds names to all the known object references. The name server provides the bound name and object references to the Jaguar server’s session manager object. Because this information is stored in memory, the name context information is retained only as long as the Jaguar server is running.

You can add persistent object name storage capabilities to Jaguar by using an external directory naming service, such as an LDAP server. The external server retains object name information, and the Jaguar name server updates this information whenever it creates new bindings or unbinds existing ones.

To use an external naming service, specify the URL of the external server in the Naming Service properties of the designated Jaguar name server. You must also provide a manager DN (distinguished name) and password that has exclusive access to all objects in the LDAP server database for Jaguar to be able to update the stored name context information.

CORBA CosNaming API support

The Jaguar naming service is an implementation of the CORBA CosNaming component. The CosNaming component is a collection of interfaces that defines the naming service. These interfaces provide support for object binding and lookup.

Jaguar implements the NamingContext interface to bind a name to an object, thereby creating a NamingContext object. Client applications use the NamingContext interface to “resolve” a bound name to its referenced object.

CosNaming::Name represents a name context that can be bound to an object implementation or another name context. CosNaming::Name is a sequence of one or more NameComponent structures. The NameComponent consists of two attributes: the identifier and the type. Both of these attributes are represented as IDL strings.

The IDL specification for NameComponent and the NamingContext interface is:

```idl
module CosNaming
  typedef string Istring;
  struct NameComponent {
    Istring id;
```
CORBA CosNaming API support

```c
Istring kind;
};
typedef sequence<NameComponent> Name;
};
```

## Binding names

There are four methods to create bindings:

- **bind** creates a binding for a name and an object to create a name context. Name contexts created using `bind` are not included when compound names are passed for resolution.

- **rebind** creates a binding for a name and an object, even if the name is already bound to an object. Name contexts created using `rebind` are not included when compound names are passed for resolution.

- **bind_context** binds a name to an existing name context. Name contexts created using `bind_context` are included when compound names are passed for resolution.

- **rebind_context** binds a name to an existing name context, even if the name is already bound to an object. Name contexts created using `rebind_context` are included when compound names are passed for resolution.

To remove an object reference from a name context, Jaguar uses the `unbind` function. When you shut down the Jaguar server, all bound objects are automatically unbound using this function. However, you can also use `unbind` when you delete a package or component from the repository. If you use persistent name storage, use `unbind` to remove references to deleted packages and components on the external server.

## Resolving Jaguar objects

Jaguar uses the `resolve` method to retrieve an object based on the name context into which it is bound. The name context used to retrieve an object must be identical to the object’s bound name context. The Jaguar naming service performs the “narrowing” of the object to the appropriate return type. In other words, the client does not need to cast the returned object to a more specialized interface.

There are two ways for Java clients to access the naming service to resolve object names:
• Using Jaguar’s CosNaming Java interface
• Using JNDI

Resolving objects using the CosNaming interface

The service provider interface (SPI) uses the CosNaming interface to connect to the Jaguar name server and retrieve the CORBA Interoperable Object Reference (IOR) associated with the server’s manager object. Once the IOR is retrieved, the naming service creates a session with the manager object and then creates an instance of the requested object. The SPI returns an instance of the requested object to the client.

After initializing the ORB, call the orb.resolve_initial_references method to obtain the initial naming context. The naming context is an object that implements the CosNaming::NamingContext IDL interface; it is used to resolve Jaguar component and service names to server-side objects.

The initial NamingContext has the name context that was specified in the com.sybase.CORBA.NameServiceURL ORB initialization property. Your client program invokes the NamingContext::resolve operation to obtain an instance of the Jaguar authentication service as well as component instances.

The NamingContext::resolve operation takes a CosNaming::Name parameter, which is a sequence of CosNaming::NameComponent structures. The Java definitions of these types and the NamingContext::resolve operation follow:

```java
package org.omg.CORBA.CosNaming;

class NameComponent {
    public String id; // Represents a node in a name
    public String kind; // Unused, can contain comments

    // Construct a NameComponent instance with the
    // specified initial values for id and kind fields
    public NameComponent(String id, String kind);
}

interface NamingContext {
    ... other methods not shown ...
    public org.omg.CORBA.Object resolve
    (NameComponent[] n)
    throws
    org.omg.CosNaming.NamingContextPackage.CannotProce,
```

53
In Java, a name is represented by an array of `NameComponent` instances, with the `id` field of each instance set to a node of the name. For example, the name:

USA/Sybase/Jaguar/TestPackage/TestComponent

can be represented by the array `theName` which is created in this code fragment:

```java
import org.omg.CORBA.CosNaming.*;
import org.omg.CORBA.CosNaming.NamingContextPackage.*;
public class myApplet extends Applet {

    NamingContext nc;
    ... deleted code that retrieves initial NamingContext ...

    NameComponent theName[] = {
        new NameComponent("USA", ""),
        new NameComponent("Sybase", ""),
        new NameComponent("Jaguar", ""),
        new NameComponent("TestPackage", ""),
        new NameComponent("TestComponent", "")
    };
}
```

For convenience, the Jaguar naming service allows you to specify multiple nodes of a name in one `NameComponent` instance, using a forward slash (/) to separate nodes. The name from the example above can be represented in a one-element array as shown below:

```java
NameComponent theName[] = {
    new NameComponent("USA/Sybase/Jaguar/TestPackage/TestComponent","")
};
```

`NamingContext::resolve` resolves a name to an object; this method either returns an `org.omg.CORBA.Object` instance or throws an exception.

For complete information about instantiating and resolving objects with CORBA naming services, see Chapter 13, “Creating CORBA-Compatible Java Clients” in the *Jaguar CTS Programmer’s Guide*. 
JNDI support

Java Naming and Directory Interface (JNDI) is a standard Java interface for accessing distributed objects and services by name. It provides a portable, unified interface for naming and directory services. The JNDI specification is independent of any specific directory or naming service such as LDAP, NDS, DCE/CDS, or NIS.

Jaguar’s JNDI implementation includes the JNDI service provider interface (SPI), which enables you to use a variety of custom directory and naming services. Jaguar uses the SPI in conjunction with the CosNaming interface to provide component lookup capability. Given a bound name, the SPI locates the referenced package and component. Once it locates the component, the SPI works with the client stub interface to instantiate the component and return the requested object.

For complete information about instantiating and resolving objects with JNDI, see Chapter 13, “Creating CORBA-Compatible Java Clients” in the Jaguar CTS Programmer’s Guide.

JNDI version level

In Jaguar servers that use JDK 1.2, the JNDI InitialContext object follows the JNDI 1.2 interface specification. In servers that use JDK 1.1, JNDI InitialContext follows the JNDI 1.1 interface specification. When you start the Jaguar server, the JNDI classes required for the server’s JDK version are configured automatically.

JNDI J2EE features

Jaguar supports the JNDI features required by the Java 2 Enterprise Edition (J2EE) platform specification. You must run the JDK 1.2 version of the Jaguar server to use these features.

In J2EE, you can use the application component’s naming environment to customize an application’s business logic without accessing the source code. The application component’s container implements the environment as a JNDI naming context and provides the JNDI interfaces to access the environment properties that you define in the deployment descriptor.
Environment properties

When you deploy a J2EE application, use the deployment descriptor to define all the environment properties that the application component needs to access. This sample code defines the environment property (env-entry) `maxExemptions` as an Integer and sets its value to 10:

```xml
<env-entry>
  <description>
    The maximum number of tax exemptions
  </description>
  <env-entry-name>maxExemptions</env-entry-name>
  <env-entry-type>java.lang.Integer</env-entry-type>
  <env-entry-value>10</env-entry-value>
</env-entry>
```

The information between the opening and closing `env-entry` tags defines an environment entry element, which consists of:

- **description** This is optional.
- **env-entry-name** The environment property name, relative to the `java:comp/env` context.
- **env-entry-type** The environment property’s Java datatype must be one of: Boolean, Byte, Double, Float, Integer, Long, Short, or String.
- **env-entry-value** The environment property value, which is optional.

Within the same container, all instances of an application component share the same environment properties. The component instances cannot modify the environment at runtime.

An application component instance uses the JNDI interfaces to locate the environment naming context and access the environment properties. To locate the naming context, an application creates a `javax.naming.InitialContext` object and gets the `InitialContext` for `java:comp/env`. In this example, the application retrieves the value of the environment property `maxExemptions` and uses that value to determine an outcome:

```java
Context initContext = new InitialContext();
Context myEnv = 
    (Context)initContext.lookup("java:comp/env");

// Get the maximum number of tax exemptions
Integer max=(Integer)myEnv.lookup("maxExemptions");

// Get the minimum number of tax exemptions
Integer min = (Integer)myEnv.lookup("minExemptions");
```
// Use these properties to customize the business logic
if (numberOfExemptions > max.intValue() ||
    (numberOfExemptions < min.intValue()))
    throw new InvalidNumberOfExemptionsException();

Default name service
When you call the empty constructor to create a new InitialContext, Jaguar sets
the Context.INITIAL_CONTEXT_FACTORY system property and sets
Jaguar’s EJB name service as the default.

For information about using Jaguar Manager to add and configure environment
properties in Web applications, see Chapter 22, “Creating Web Applications”
in the Jaguar CTS Programmer’s Guide.

EJB references
An EJB reference identifies the home of an enterprise Bean. You can use the
deployment descriptor to create a link between an EJB reference and an
enterprise Bean, contained within an EJB JAR file. Deployment descriptor
interfaces allow an application component to access an enterprise Bean’s home
interface using EJB references.

To locate an enterprise Bean’s home interface, declare an EJB reference in the
deployment descriptor and use JNDI to look up the interface. The referenced
enterprise Bean must be in the ejb subcontext of the application component’s
environment.

You can declare an EJB reference in the deployment descriptor using the ejb-ref
element. The data between the opening and closing ejb-ref tags defines an
ejb-ref element. This code sample defines an EJB reference to the Employee
text Bean:

    <ejb-ref>
        <description>
            Reference to the Employee entity Bean
        </description>
        <ejb-ref-name>ejb/Employee</ejb-ref-name>
        <ejb-ref-type>Entity</ejb-ref-type>
        <home>com.wooster.empl.EmployeeHome</home>
        <remote>com.wooster.empl.Employee</remote>
    </ejb-ref>

An ejb-ref element contains:
**JNDI support**

- **description**  This is optional.
- **ejb-ref-name**  The name of the Bean used in the application component.
- **ejb-ref-type**  The Bean type, Entity or Session.
- **home**  The expected Java type of the home interface.
- **remote**  The expected Java type of the remote interface.
- **ejb-link**  This is optional.

This code sample illustrates how to use JNDI to look up the home interface of the *Employee* enterprise Bean:

```java
// Get the default initial JNDI context
Context initContext = new InitialContext();

// Look up the home interface of the Employee enterprise Bean
// Object result =
//   initContext.lookup("java:comp/env/ejb/Employee");

// Convert the result to the correct type
EmployeeHome empHome = (EmployeeHome)
   javax.rmi.PortableRemoteObject.narrow(result, EmployeeHome.class);
```

**Declaring an EJB link**

You can define a link from an EJB reference to an enterprise Bean by declaring an **ejb-link** element in the deployment descriptor. The application component and the target enterprise Bean must be in the same J2EE application. This sample code creates a link to the *Employee* enterprise Bean, by adding an **ejb-link** element to the Bean’s EJB reference definition:

```xml
<ejb-ref>
  <description>
    Reference to the Employee entity Bean
  </description>
  <ejb-ref-name>ejb/Employee</ejb-ref-name>
  <ejb-ref-type>Entity</ejb-ref-type>
  <home>com.wooster.empl.EmployeeHome</home>
  <remote>com.wooster.empl.Employee</remote>
  <ejb-link>Employee</ejb-link>
</ejb-ref>
```

For information about using Jaguar Manager to add and configure EJB references in Web applications, see Chapter 22, “Creating Web Applications” in the *Jaguar CTS Programmer’s Guide*. 

---

58
For information about using Jaguar Manager to add and configure EJB
references in EJB components, see Chapter 8, “Creating Enterprise JavaBean
Components” in the *Jaguar CTS Programmer’s Guide*.

**Resource factory references**

A resource factory is an object that you use to create resources. You can assign
a logical name to a resource factory in the deployment descriptor.

A *resource-ref* element defines a single resource factory reference. This code
sample defines a reference to the resource factory that implements the
*DataSource* interface:

```xml
<resource-ref>
    <description>
        Data source for the database in which the Employee enterprise Bean records transactions
    </description>
    <res-ref-name>jdbc/EmployeeAppDB</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
</resource-ref>
```

A *resource-ref* element contains:

- **description**  This is optional.
- **res-ref-name**  Resource reference name used in the application’s code.
- **res-type**  Resource Java datatype that the application expects.
- **res-auth**  Resource sign-on authorization, Bean or Container.

This code sample obtains a reference to the resource factory that implements
the *DataSource* interface, and uses that reference to get a database connection
(resource):

```java
// Obtain the initial JNDI context
Context initContext = new InitialContext();

// Look up the resource factory using JNDI
javax.sql.DataSource ds = (javax.sql.DataSource) initContext.lookup
    (“java:comp/env/jdbc/EmployeeAppDB”);

// Get a database connection
java.sql.Connection connection = ds.getConnection();
```
Configuring the Jaguar naming service

For information about using Jaguar Manager to add and configure resource references in Web applications, see Chapter 22, “Creating Web Applications” in the Jaguar CTS Programmer’s Guide.

For information about using Jaguar Manager to add and configure resource references in EJB components, see Chapter 8, “Creating Enterprise JavaBean Components” in the Jaguar CTS Programmer’s Guide.

UserTransaction references

J2EE application components can use the Java Transaction API (JTA) UserTransaction interface to manage transactions. A component instance can look up an object that implements the interface using the JNDI name java:comp/UserTransaction.

In this code sample, an application component uses the interface to manage a transaction:

```java
// Get the initial JNDI context
Context initContext = new InitialContext();

// Look up the UserTransaction object
UserTransaction tran = (UserTransaction)
    initContext.lookup("java:comp/UserTransaction");

// Start a transaction
tran.begin();

// data updates

// Commit the transaction
tran.commit();
```

Configuring the Jaguar naming service

Use the Naming Service tab on the Server Properties window to set the Jaguar server’s naming service options. You can use the Naming Service properties to configure a Jaguar server to be a name server, or point to another Jaguar server as its name server.

The Naming Service property sheet includes:

- The Jaguar server’s initial context.
• Whether or not the Jaguar server is enabled as a name server.
• If the server is not enabled as a name server, the URL for the Jaguar server acting as the name server.
• Heartbeat detection – periodically verifies that clustered name servers are either accepting client connections or have failed. See “Heartbeat detection” on page 70 for more information.
• If you are using an LDAP server to provide persistent name storage, the URL of the LDAP name server, as well as the manager DN (distinguished name) for the LDAP server.

For complete information about setting the Naming Service properties for a Jaguar server, see “Naming Service” on page 11.

Name binding password security

You can establish password protection on the Jaguar naming service to allow name binding only from designated Jaguar servers. This prevents unauthorized applications from creating name bindings using a Jaguar name server.

To use the name binding password feature, you must set the:

```
com.sybase.jaguar.server.CosNaming.bindpassword
```

property for the name server and each server participating in the naming service. You set this property using the All Properties tab in the Server Properties window. The default value is “jaguar.”

All servers participating in the password-protected name service must have the same password as the name server. If the bindpassword property is empty, or does not exist in the property file for a name server, the name server accepts binds from any source.

Using an LDAP server with Jaguar

To add persistent object name storage capabilities to Jaguar, you can use an external directory naming service, such as an LDAP server. The Jaguar server properties include an optional URL for specifying the port for the external name server.
Using an LDAP server with Jaguar

When you use an external name server, Jaguar uses JNDI to communicate with the name server through the specified URL.

LDAP object schema and Jaguar objects

LDAP servers have predefined schema for common objects such as country, organization, and organizational unit. Jaguar uses the following format for an LDAP-compatible initial context:

\[ ou=\text{organizational unit}, o=\text{organization}, c=\text{country} \]

Storing Jaguar object bindings on an LDAP server

When you use an LDAP server with a Jaguar name server, the CosNaming component binds all implemented objects on the servers that use the designated Jaguar name server, and stores the name context information on the LDAP server. If Jaguar detects previously-bound objects on the external name server, it updates the existing bindings with current name context information. When you shut down the Jaguar server, it unbinds the stored information.

❖ To connect a Jaguar name server to an LDAP server:

1. On start-up, the Jaguar name server connects to the LDAP server using the URL specified in the Jaguar name server’s Naming Service properties.
2. The Jaguar name server authenticates the connection to the LDAP server using the manager DN specified in the Jaguar name server’s Naming Service properties.
3. The Jaguar name server attempts to retrieve any existing matching name contexts from the LDAP server. If successful, the Jaguar name server uses the existing name context information.
4. The Jaguar name server prepares the server object with the required attributes.
5. The Jaguar name server attempts to add the server object to the LDAP server. If the object already exists, the LDAP server updates the existing object with the current attributes.
6. The Jaguar server adds any new package/component name context information, or modifies the existing information if necessary.
CHAPTER 3

A cluster is a group of servers that share replicated repository information. A Jaguar cluster’s primary purpose is to provide load balancing and high availability. See Chapter 4, "Load Balancing, Failover, and Component Availability" for more information about load balancing and high availability. Synchronization enables you to connect to the primary Jaguar server in a cluster and distribute repository information to “synchronize” one or more of the other server(s) in the cluster. You can also synchronize non-clustered Jaguar servers. Synchronization provides a quick and easy way to distribute package, servlet, and other configuration information between servers.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster overview</td>
<td>63</td>
</tr>
<tr>
<td>Configuring a cluster</td>
<td>66</td>
</tr>
<tr>
<td>Synchronization</td>
<td>73</td>
</tr>
</tbody>
</table>

Cluster overview

Each cluster includes a primary server, a group of participating servers, and a set of name servers:

- The primary server contains the master copy of the configuration repository for all servers in the cluster. The primary server distributes (synchronizes) its configuration to the other servers in the cluster.

- Participating servers or non-primary servers share a “logical server name,” which corresponds to a Jaguar server defined in the primary server’s repository. Several physical servers in a cluster share a logical server name; each like-named server shares components and servlets, and uses the same connection caches and other configuration information.
Cluster overview

When you are configuring a cluster, you can use multiple logical server names to partition components. To ensure high availability, each logical server name must be shared by at least two physical servers in the cluster.

**Note** All servers within a cluster can share the same name as long as you are not partitioning.

- A cluster consists of at least two *name servers*. Each Jaguar server in a cluster is aware of all of the name servers. Each Jaguar server in a cluster binds its components to all name servers in the cluster. Binding all of the components of the clustered Jaguar servers to multiple name servers provides high availability of your business components and redundancy if a Jaguar server should go down, even if it is a name server.

The list of name servers for the cluster override the naming properties for participating logical server definitions.

Cluster support is tightly integrated with the Jaguar naming service, so that all client services from a cluster are made available through the naming service. See Chapter 2, “Jaguar Naming Services” for more information.

Typically, each server in a cluster runs on a different host, so each server has its own copy of the entire repository and all files required for component execution. Sybase recommends that you run each cluster member from its own installation directory.

Figure 3-1 illustrates a Jaguar cluster and uses hostA as the primary server to synchronize the participating servers, including the name servers. Each server in the cluster is named “Jaguar.”
All hosts of a cluster must be of the same platform type; in other words, you cannot have NT and UNIX machines in the same cluster.

**Note** The SessionManager::Session::lookup operation implicitly consults the naming service, so you can write a client that does not explicitly use the naming service but still takes advantage of cluster services. For example, PowerBuilder’s connection object uses SessionManager::Session::lookup.

**Cluster servers**

Each server can be a member of only one cluster. To provide high availability, there should be at least two name servers defined for a cluster.

Servers are defined by URL, rather than by Jaguar server name. Every server in a cluster can be named Jaguar. If you are not using partitioning, it is easy to add a new machine to a cluster; simply change “localhost” in each listener to the Internet host name or IP address. Then, connect Jaguar Manager to the cluster primary and synchronize the cluster.
A name server can be used by one or more clusters. See “Adding a name server to a cluster” on page 68 for restrictions before you assign a name server to more than one cluster.

All servers in a cluster should share a common account that will be used for inter-server connections when synchronizing the cluster. This must be a jagadmin account or an account for a user that has the Admin role.

If the servers in a cluster do not share a common account, you can synchronize the cluster only on a server-by-server basis.

Configuring a cluster

This section describes the steps required to create, configure, and manage a cluster using Jaguar Manager.

❖ Creating a cluster from Jaguar Manager

1 Highlight the Clusters folder.
2 Select File | New Cluster.
3 Enter the name of the new cluster in the New Cluster dialog box and click Create New Cluster.

The names of all clusters appear on the right side of the window when you highlight the Clusters folder.

To set the initial context used by servers in a cluster when they bind their objects into the name servers, set the com.sybase.jaguar.cluster.initialcontext property on the All Properties tab in the Cluster Properties dialog. For example, cluster “US_Cluster” may have initial context “/US”, and cluster “UK_Cluster” may have initial context “/UK”. Then clients can talk to any name server which is used by either “US_Cluster” or “UK_Cluster”, and by specifying the appropriate initial context, (either “/US” or “/UK”) the clients can be directed to the appropriate servers.

❖ Configuring a server to enable synchronization

Before you can add a server to a cluster, you must configure each physical server to enable synchronization from the primary server.
1 Use Jaguar Manager to connect to the physical server. If necessary, define a server that matches a logical server name defined in the primary server’s repository. In other words, if the primary server’s name is Jaguar_cluster, you must also assign the name Jaguar_cluster to each participating server.

2 If necessary, configure the listeners of this server. You must at least configure an IIOP listener to match the URL that will be used for cluster synchronization. Each like-named server in a cluster must also share the same listener name.

3 Configure the account that will be used to synchronize the cluster.
   • If you are using jagadmin, change the jagadmin password to match that of the primary server.
   • If you are using an account other than jagadmin, add that account’s user name or digital ID to the Admin role.

4 Restart the server so the new network addresses and security changes take effect.

❖ Adding a server to a cluster
   Connect to the primary server with Jaguar Manager and add each physical server to the cluster definition as follows:
   1 Click the Clusters icon.
   2 Select the cluster to which you want to add servers.
   3 Select File | Cluster Properties. The primary server (the server to which you are connected) displays in the Primary Server field of the General tab.
   4 Select the Servers tab.
   5 Click Add.
   6 Enter a valid IIOP address for the server, for example iiop://myhost:9000. This address is used for inter-server connections when you synchronize the cluster.
7 Once you have added all the servers that you intend to add to the cluster, synchronize the cluster. The Synchronize dialog box appears automatically after you add a server and click OK. See “Synchronizing a cluster from Jaguar Manager” on page 77 for more information. You can add more servers later, but you must then again synchronize the cluster.

---

Warning! After you add a non-primary server to a cluster, Jaguar Manager warns you when you connect directly to that server. Direct user updates to the server’s configuration can be overwritten when the cluster is synchronized if the new server has been the target of at least one synchronization before it was added as a member of the cluster.

❖ Adding a name server to a cluster

1 Click the Clusters icon.
2 Select the cluster to which you want to add name servers.
3 Select File | Cluster Properties.
4 Select the Name Servers tab.
5 Click Add.
6 Enter the server’s URL (for example, iiop://myhost:9000). When you have entered all of the name servers for the cluster, click OK. The Synchronize dialog box appears automatically after you add a name server and click OK. You can add more name servers later but must synchronize the cluster afterwards.
7 Synchronize the cluster.

---

Note Name servers can also be “ordinary” members of a cluster (if they are also listed on the Servers tab in the Cluster Properties dialog). However, if a name server is used by more than one cluster, it can be an ordinary member of only one cluster.

Jaguar requires the cluster’s bind password to authorize name context updates to the cluster’s name servers. When you create a cluster, a random bind password is automatically generated. In most cases, you do not need to edit the password. However, if a name server is used by two or more clusters, you must configure the clusters to use the same bind password.
To change the password, modify the
\texttt{com.sybase.jaguar.cluster.bindpassword} property on the All
Properties tab in the Cluster Properties dialog. Sybase recommends that you
use one of the randomly generated passwords, as security can be compromised
if clients obtain knowledge of a cluster’s bind password.

\begin{itemize}
\item[-] \textbf{Removing a server from a cluster}
\begin{enumerate}
\item Click the Clusters icon.
\item Select the cluster from which to remove the server.
\item Select File | Cluster Properties.
\item Select the Servers tab.
\item Highlight the server you want to delete and click Delete.
\item Connect to the deleted server using Jaguar Manager, and delete the
\texttt{com.sybase.jaguar.server.cluster} property from the Server
Properties dialog (this tells the server it is no longer a member of the
cluster):
\begin{enumerate}
\item Double-click the Servers folder.
\item Highlight the server that has been deleted from the cluster.
\item Select File Server Properties.
\item Select the All Properties tab.
\item Highlight the \texttt{com.sybase.jaguar.server.cluster} property.
\item Click Delete.
\end{enumerate}
\item Synchronize the cluster, and restart the name servers.
\end{enumerate}
\end{itemize}

\begin{itemize}
\item[-] \textbf{Removing a name server from a cluster}
\begin{enumerate}
\item Click the Clusters icon.
\item Select the cluster from which to remove the name server.
\item Select File | Cluster Properties.
\item Select the Name Servers tab.
\item Highlight the server you want to delete and click Delete.
\item Synchronize the cluster.
\end{enumerate}
\end{itemize}

\begin{itemize}
\item[-] \textbf{Deleting an existing cluster}
\begin{enumerate}
\item Click the Clusters icon.
\end{enumerate}
\end{itemize}
Configuring a cluster

2 Select the cluster you want to delete.
3 Select File | Delete Cluster.

A server that was part of the deleted cluster may try to connect to the cluster. In these cases you see error messages in the *srv.log* file indicating that the server is in Admin mode and instructing you to synchronize the cluster to fix the problem. To do so:

1 Highlight the server.
2 Select File | Server Properties.
3 Select the All Properties tab.
4 Delete the *com.sybase.jaguar.server.cluster* property.

❖ Rebinding a cluster

The Rebind option refreshes all of the name servers within a cluster. If you add a component to a server that is already part of a cluster and want to make that component available to the cluster, you need to rebind the cluster. You can also use the rebind option if a problem occurs when you synchronize the cluster; if for example, one of the name servers is slow to start.

1 Highlight the name of the cluster.
2 Select File | Rebind.

Heartbeat detection

The name servers in a cluster use heartbeat detection to periodically verify that member servers are either accepting client connections or have failed. If a server is not accepting connections, the name server does not return profile (host:port) information to the client, and routes requests to other servers in the cluster. The name server also detects when a failed server is ready to accept connections again and starts routing client requests to that server.

If a name server using transient storage fails, the cluster rebinds automatically when you reboot the failed name server. Otherwise, the cluster provides access to components through the remaining name servers in the cluster.

If a name server using persistent storage and LDAP support fails, the cluster does not need to rebind, but LDAP may leave behind stale profiles resulting in unnecessary client retries and failures. For this reason, Sybase recommends you use transient storage to support load balancing and high availability.
❖ To enable heartbeat detection from Jaguar Manager:
1. Select the Servers folder.
2. Highlight the name server for which you are configuring heartbeat detection.
4. Open the Naming Service tab.
5. Select the Enable Heartbeat check box.
6. Enter the heartbeat frequency. This number is how often, in seconds, that the name server checks server availability. As the frequency period is shortened, server performance decreases. The default frequency is 120 seconds.

When you synchronize a cluster, the heartbeat settings (whether or not it is enabled and frequency) of the primary server are distributed to the other name servers in the cluster.

The property `com.sybase.jaguar.cluster.primary` stores the primary server URL for each cluster. The synchronization process maintains this automatically; you cannot manually change it using the All Properties tab. The cluster property `com.sybase.jaguar.cluster.version` stores the version number for each cluster. The synchronization process maintains this automatically.

❖ To view the load on a cluster from Jaguar Manager:
1. Select the Clusters folder.
2. Highlight the cluster you want to view.

The load graph is a bar chart. The load of each server is expressed as a percentage. Load is defined as the ratio of the number of bytes sent in the last 1/2 second over 10,000, expressed as a percentage.

The status list contains an entry for each server and tells whether the server is in Ready or Admin mode. If a server is in Admin mode, the list tells why.
Cluster start-up options

On start-up, a Jaguar server that is a member of a cluster uses the com.sybase.jaguar.cluster.startup cluster property to determine whether it is “in sync” with other cluster members. Check or modify this property from the All Properties tab in the Cluster Properties window.

com.sybase.jaguar.cluster.startup can have these values:

- **check_primary** (default value)
  a. If any other cluster member (including the primary) is reachable and has a higher cluster version number, this server moves to Admin mode since it is not “in sync”.
  b. If the primary is available and has the same cluster version number, this server joins the cluster and is ready to accept client connections.
  c. If the primary is available and has a different cluster version number, this server moves to Admin mode since it is not “in sync”.
  d. If none of the above conditions are met, the server waits briefly then returns to step a.

- **check_servers**
  a. If any other cluster member (including the primary) is reachable and has a higher cluster version number, this server moves to Admin mode since it is not “in sync”.
  b. Let $M$ be the number of cluster members (including the primary), and let $N$ equal $M/2$ (integer division). If at least $N$ other servers are available and have the same cluster version number, this server joins the cluster and is ready to accept client connections.
  c. If none of the above conditions are met, the server waits briefly then returns to step a.

The check_servers option allows a server to join a cluster even if the primary is not available. This prevents a single point of failure if the primary server becomes unavailable.

- **disable_check** – no checking is performed. Sybase discourages the use of this option since it can result in “out-of-sync” cluster members running together.
If a server is in Admin mode due to a cluster version number mismatch, then using Set Ready is the manual equivalent of disable_check, so using Set Ready in this case is also discouraged as it may result in “out-of-sync” cluster members running together. Sybase recommends that you synchronize the cluster instead. See “Synchronizing a cluster from Jaguar Manager” on page 77 for more information.

A server in Admin mode can be put into ready mode using the Set Ready option. See “Synchronizing a cluster from Jaguar Manager” on page 77 for more information.

**Synchronization**

Synchronization replicates application files and configuration information between Jaguar servers. If you are using clusters, synchronization ensures that logical servers in a cluster share the same application files and configuration. Synchronization is also a useful alternative to importing and exporting packages.

For example, as Figure 3-2 illustrates, you can replicate new components from a testing or development server to one or more production servers. As long as you can connect to both servers, synchronization is quicker and easier than exporting and importing package files.
If you are using clusters and make configuration changes to the primary server, jagadmin (or any user with the Admin role) can synchronize those changes to participating non-primary servers.

To synchronize a cluster, you must be connected to the primary server for the cluster unless the primary server is down and cannot be restarted. In this case, you can connect to another server within the cluster and designate it as the new primary server.

You can use the synchronization option to move repository information between servers that are not part of a cluster. For example, you may want to move a development Jaguar server’s repository to a test server. In this case, the synchronization command is similar to Jaguar’s export/import options without the JAR file; instead of moving packages from one server to another, you are moving repository information.

You can synchronize at the cluster, server, servlet, package, or component level. The level you choose determines the available synchronization options. Options that are not available are dimmed, and cannot be selected.
Component synchronization

The items that are synchronized for a component are the same in a Java archive (JAR) file, as for a package when using Jaguar’s Export feature, and include:

- Component definition (component properties)
- Interface definition(s)
- For C/C++ components, the DLL (or shared library) indicated by the `com.sybase.jaguar.component.cpp.library` property
- For Java components, the implementation class (property `com.sybase.jaguar.component.java.class`), and any classes referred to by the `com.sybase.jaguar.component.java.classes` property
- For PowerBuilder components, the libraries starting with a “$” that are referenced by the `com.sybase.jaguar.component.pb.librarylist` property
- Other files indicated by the `com.sybase.jaguar.component.files` property, which is a comma-separated list of any of the following:
  - Java classes (for example, `com.acme.MyClass`).
  - Java packages (for example, `com.acme`).
  - DLLs (or shared libraries), which are found in `$Jaguar/cpplib` (for example, `MySharedLibrary.so` on Solaris, AIX, Digital UNIX, and LINUX; or `MySharedLibrary.sl` on HP-UX).
  - Files with relative path names, located relative to `$Jaguar/Repository` (for example, `../dll/debug/MyDebugLibrary.dll`).
  - Files with an absolute path name, located anywhere on the host file system (for example, `d:\mydir\myfile.ext` or `/mydir/myfile.ext`). Files with absolute path names should be listed here only if all the server(s) have a file system with the same directory structure.

Note When you synchronize DLLs (or shared libraries), remote servers must all run on the same operating system and hardware. If you are synchronizing Java and/or PowerBuilder components, this restriction does not apply.
Package synchronization

When you synchronize a Jaguar package, these items are distributed to the remote server(s):

- The same items as for component synchronization for each component in the package.
- Other files indicated by property `com.sybase.jaguar.package.files`. This property is interpreted in the same way as `com.sybase.jaguar.component.files`.

See “Component synchronization” on page 75 for more information.

Servlet synchronization

When you synchronize a Jaguar servlet, the following items are distributed to the remote server(s):

- The servlet definition (servlet properties).
- The servlet class (`com.sybase.jaguar.servlet.java.class` property), and any classes referred to by the `com.sybase.jaguar.servlet.java.classes` property.
- For PowerBuilder components, the libraries starting with a “$” that are referenced by the `com.sybase.jaguar.component.pb.librarylist` property.
- Other files indicated by property `com.sybase.jaguar.servlet.files`. This property is interpreted in the same way as `com.sybase.jaguar.component.files`.

See “Component synchronization” on page 75 for more information.

Application synchronization

When you synchronize a Jaguar application, the following items are distributed to the remote server(s):

- The application definition (application properties)
- The packages, Web applications, and the other resources contained in the application
Web application synchronization

When you synchronize a Jaguar Web application, the following items are distributed to the remote server(s):

- The Web application definition (Web application properties)
- The servlets, JSPs, and other files contained in the Web application

❖ **Synchronizing a cluster from Jaguar Manager**

1. Use Jaguar Manager to connect to the source (primary) server.
2. Depending on the level of the sync operation, highlight a cluster, server, package, servlet, or component.
4. Supply the synchronization information in the Synchronization dialog and click Start Sync. Table 3-1 describes the synchronization properties.

A server is in Admin mode if it is running and accepting connections from Jaguar Manager, but not accepting ordinary client connections. A server in Admin mode can be the target of synchronization, since this is often the appropriate way to get it out of Admin mode. Jaguar Manager warns you if you connect to a server that is running in Admin mode. You can use the “Set Ready” option in Jaguar Manager to force a server into Ready mode, but you should understand why the server is in Admin mode before you use this option.

❖ **Overriding synchronization operations**

You may want to manually configure properties on individual servers in a cluster to override properties from the logical server definition. For example, you may want to:

1. Increase the size of a connection cache on a machine that has more memory than other machines.
2. Establish a process for making manual updates after each Jaguar Manager synchronization.

### Table 3-1: Synchronization properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>The Jaguar user name used to log in to the remote server(s).</td>
<td>The user must belong to the Admin Role to use the synchronize option.</td>
</tr>
</tbody>
</table>
## Synchronization

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>The password for the Jaguar user name on the remote server(s).</td>
<td>When synchronizing to multiple remote servers, including cluster members, all remote servers must use the same password to allow connection to those servers.</td>
</tr>
<tr>
<td>Cluster</td>
<td>Click the Cluster check box and select a cluster from the drop-down list to synchronize the entire cluster.</td>
<td>If you are synchronizing a server, package, component, or servlets, you can specify that the changes be replicated to all servers in a cluster. Use this option only when connected to the cluster’s primary server, or when connected to a non-primary server that will become the primary after synchronization.</td>
</tr>
<tr>
<td>Servers</td>
<td>Click the Servers check box and supply a list of one or more URLs to be targets of synchronization if you want to synchronize a subset of the cluster.</td>
<td>The URLs must be separated by a comma, and of the form <code>iiop://hostname:port</code>.</td>
</tr>
</tbody>
</table>
| All Cluster Files | When this option is selected, all of the cluster files are distributed to the remote server(s). Cluster files contain a set of properties needed to replicate repository information, including:  
• `afconfig.dat` – includes some server, role, and connection cache definitions  
• All security profiles  
• The Jaguar /Repository/Security directory |                                                                                                                                                  |
| All Package Files | When this option is selected, all of the package files are distributed to the remote server(s). Package files refers to all user-defined IDL files, property files, and implementation files (Java .class, .dll, .so, .pbd) based on component definitions. | Selecting this option may result in the synchronization taking a long time, depending on the number of files being transferred to the remote server(s). Consider using component level synchronization if only a few components have been changed. See “Component synchronization” on page 75 and “Package synchronization” on page 76 for more information. |
| All Servlet Files | When this option is selected, all servlet files are distributed to the remote server(s) | See “Servlet synchronization” on page 76 for more information.                                                                                     |
| All App Files   | When this option is selected, all application files are distributed to the remote server(s) | See “Application synchronization” on page 76 for more information.                                                                                   |
| All Web App Files | When this option is selected, all Web application files are distributed to the remote server(s) | See “Web application synchronization” on page 77 for more information.                                                                           |
Chapter 3  Jaguar Clusters and Synchronization

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verbose</td>
<td>Select this check box if you want maximum information regarding the progress of the synchronization sent to the srv.log file of the primary server.</td>
<td>This option is only appropriate when synchronizing a cluster. Note This option is intended as a safeguard to prevent synchronization of a cluster from the wrong source and should be used only if the source has changed.</td>
</tr>
<tr>
<td>New Primary</td>
<td>When a cluster is synchronized, the primary is defined as the URL of the server from which the synchronization was initiated. Subsequently, if you want to synchronize the cluster from a different source, you must select the New Primary option in the Synchronize dialog.</td>
<td></td>
</tr>
<tr>
<td>New Version</td>
<td>When this option is selected, a new cluster version number is generated.</td>
<td>When a cluster server restarts, it checks if the cluster version number in its local repository matches the version number in the primary server's repository. If not, the server does not allow ordinary client connections until the situation is rectified (the server in this case is said to be “out of sync”). If the primary server is not available, the restarting server waits until the primary becomes available. If other problems are found with the cluster definition, or a previous synchronization failed, the server stays in Admin mode. You may want to keep the primary server in Admin mode all of the time so clients never run components on it, and it becomes a staging area for changes. This option is appropriate only when you are synchronizing a cluster.</td>
</tr>
<tr>
<td>Refresh</td>
<td>This option specifies that a component, package, or server level refresh be performed at the end of synchronization. This has the same effect as using Jaguar Manager to connect to each of the remote servers and performing a refresh at the respective level.</td>
<td></td>
</tr>
<tr>
<td>Restart</td>
<td>If you select this option, the servers you are synchronizing restart when synchronization completes.</td>
<td>A time-out is used so that if a remote server does not restart, the synchronization moves to the next server.</td>
</tr>
<tr>
<td>Then Wait</td>
<td>This is the amount of time that the primary waits for a server to restart if you have selected the Restart option.</td>
<td></td>
</tr>
</tbody>
</table>
Note  Synchronization is enabled for servers, packages, and components for any Jaguar server edition. Cluster synchronization is enabled only for the Enterprise edition. The cluster options show up in the Synchronize dialog for any object, but are disabled if the Enterprise edition is not licensed.

❖ To switch a Jaguar server from Admin mode to Ready mode:
1. Double-click the Servers icon.
2. Highlight the server for which you want to change the mode.
Load Balancing, Failover, and Component Availability

This chapter discusses:

- Load balancing – optimizes performance for your Jaguar cluster by adjusting the load across the servers.

- Component deployment – you can restrict access to components by deploying them to a subset of servers within a cluster, or make them available from all servers.

- High availability – a Jaguar cluster provides redundancy (high availability) of business components and Jaguar services in case a server within a cluster fails.

- Automatic component failover – allows a client’s object reference to be usable across servers should a server within a cluster fail.

- Sybase Failover for high availability systems – you can implement the failover feature in Adaptive Server Enterprise 12.0 with Jaguar server database connectivity using Java Connection Management (JCM).

To enable these features, you must first create a server cluster (a group of Jaguar servers running on different machines). The servers in a cluster share the workload and provide client services even if one or more servers within the cluster fails or is offline. See Chapter 3, “Jaguar Clusters and Synchronization” for information about creating a cluster.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load balancing overview</td>
<td>82</td>
</tr>
<tr>
<td>Configuring load balancing</td>
<td>85</td>
</tr>
<tr>
<td>Component deployment</td>
<td>86</td>
</tr>
<tr>
<td>High availability</td>
<td>87</td>
</tr>
<tr>
<td>Automatic failover</td>
<td>88</td>
</tr>
<tr>
<td>Implementing Sybase Failover for high availability systems</td>
<td>91</td>
</tr>
</tbody>
</table>
Load balancing overview

Load balancing in a Jaguar cluster is determined by three factors, each of which is discussed in detail in this section:

- Load metrics – if you select a dynamic load policy, the load metrics determine the load on your servers and give each server a numerical weighting, which is then used to distribute incoming client requests and optimize performance of the cluster.

- Load distribution policy – when you configure load balancing, you select the distribution policy that best matches your environment and situation.

- Interoperable object reference (IOR) – contains a profile that the client uses to look up a component. The profile contains the server and port number that the client uses to access the component. The distribution policy determines the order in which the profiles are distributed to the clients.

Load metrics

Load metrics is a collection of system statistics that define a load on a Jaguar server. Each server in a cluster is assigned a value, or normalized load, based on the load metrics.

Many factors affect overall system performance and throughput of a Jaguar server. The load metrics that Jaguar uses to determine the normalized load are:

- CPU utilization
- Method response time
- IIOP connections

When overall cluster load is light, incoming requests are evenly distributed to all member servers; that is, all member servers have some load. When the cluster becomes more loaded, the load is distributed according to a server’s current load.
Various system load metrics are collected at each sampling interval by the load collector in each Jaguar server. At each broadcasting interval, the load collector broadcasts its load metrics to all member servers. At each calculating interval the load collectors in the name servers calculate and generate a normalized load list (NLL) of all member Jaguar servers sorted according to their load metrics. The NLL consists of two elements: the server name and normalized load number. For example, a three-server cluster may have a NLL of:

(Jaguar_server_A, 5), (Jaguar_server_B, 3), (Jaguar_server_C, 1)

Which indicates that the load of Jaguar_server_A is five times greater than the least-loaded server, Jaguar_server_C.

A normalized load number ranges from 1 to 5 and is generated by combining all weighted load metrics. A value of 5 indicates a heavily loaded server, while a 1 indicates a lightly loaded server.

All name servers have their own copies of the NLL, which reflects the Naming Service’s view of system-wide load distribution that remains static for an entire refresh interval.

The NLL is referenced to balance the load of a cluster if an adaptive load distribution policy is used. See “Load distribution policies” on page 83 for information about the various distribution policies.

Load distribution policies

There are four load distribution policies:
• Random – static, even distribution.
• Round-robin – static, even distribution.
• Weighted – static, random based on server weight.
• Adaptive – dynamic, random based on load metrics.

The first three policies do not rely on load metrics, so there is no need to obtain an NLL.

The naming service applies the selected load distribution policy and generates an IOR with multiple profiles. See “Interoperable object references” on page 84 for more information.
Load balancing overview

Random and round-robin policies
Both random and round-robin policies attempt to evenly distribute incoming client requests to all participating Jaguar servers. These distribution policies are ideal for a cluster with comparable physical nodes that have similar performance characteristics.

Weighted policy
The weighted policy allows you to specify the processing load to each server. This policy is designed for physical nodes with wide range of performance variations. You can specify any weight between 1 and 10 for each server. A larger number indicates that the server can handle a larger load.

Adaptive policy
The adaptive policy is similar to the weighted policy but adapts for runtime load variation. It examines the NLL for current load metrics, and compensates for uneven loads by distributing proportionately more client requests to lightly loaded servers. There is more overhead with the adaptive policy than the other policies.

Load collectors reside in each Jaguar server, and the failure of any load collector has no global effect. If a broadcast of load metrics from a particular load collector has not been received for a predefined period of time, the corresponding server is dropped from the NLL.

The load distribution is fully synchronized and is linked to the name servers as part of the naming services. The high availability protection for naming services also applies to dynamic load balancing. See “High availability” on page 87 for more information.

Interoperable object references
Load balancing uses Jaguar’s naming service to distribute incoming IIOP requests across the servers within a Jaguar cluster:

1. The client obtains a factory IOR from the name server when it performs a lookup operation on a component. This factory IOR contains a profile (a server:port combination) that identifies the servers from which the component is available. There is a profile for each server, or if a single server has multiple IIOP ports, a profile for each port.

   The name server uses the selected load distribution policy and generates an IOR with multiple profiles to balance the requests between available servers and ports. If a dynamic load policy is selected, the NLL is used to determine and balance the load of the individual servers.

2. Using the factory IOR, the client contacts a server using one of the profiles to obtain the IOR for the component. The IOR for the component has only one profile within it.
3 The client sends the IIOP request to the profile specified by the IOR of the component.

See the Jaguar CTS Programmer's Guide for detailed information.

**Configuring load balancing**

To enable and configure load balancing from Jaguar Manager:

1 Select the Clusters folder.

2 Select the cluster that you want to configure.

3 Select File | Cluster Properties. The Cluster Properties window displays.

4 Select the Load Balancing tab. Click the Dynamic Load Balancing Enabled check box.

5 Complete the load balancing property sheet and click OK. Default values are provided but you may want to modify them depending on the capabilities of the servers in your cluster. Load balancing properties are described in Table 4-1.
In addition to load balancing to distribute incoming client requests, you can make your components available on every server within your cluster or restrict access to components by partitioning your applications.

To deploy components on a cluster so that every component is available from every cluster member, see Chapter 3, “Jaguar Clusters and Synchronization”.

Table 4-1: Load balancing properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample interval</td>
<td>The interval, in seconds, that each server collects its load metrics.</td>
<td>The default value is five seconds.</td>
</tr>
<tr>
<td>Broadcast</td>
<td>The interval, in minutes, that each server broadcasts its load metrics to the other servers in the cluster.</td>
<td>The default value is five minutes.</td>
</tr>
<tr>
<td>Calculate</td>
<td>The interval, in minutes, that the load collectors in the name servers calculate and generate a normalized load list (NLL) of all member Jaguar servers.</td>
<td>The default value is ten minutes.</td>
</tr>
<tr>
<td>Refresh interval</td>
<td>The interval, in minutes, that the name servers obtain the NLL from its local load collector. All name servers have their own copies of the NLL. The refresh interval is equal to or greater than the calculate interval.</td>
<td>The default value is ten minutes.</td>
</tr>
<tr>
<td>Maximum weight</td>
<td>The maximum weight, used in a weighted load balancing policy, of any server.</td>
<td>The default maximum weight is five. The maximum value is ten.</td>
</tr>
<tr>
<td>Policy</td>
<td>Select the load balancing policy for this cluster from the drop-down list.</td>
<td>If you select a weighted policy, you must supply a weighted value for each server in the cluster.</td>
</tr>
<tr>
<td>Weights</td>
<td>This property is enabled if you selected a weighted policy. Supply a weighted value for each server in the cluster, not to exceed the maximum weight value.</td>
<td>See “Weighted policy” on page 84 for more information.</td>
</tr>
</tbody>
</table>
Partitioning restricts components to a subset of servers within a cluster. Partitioning an application by creating different logical server names, then installing component subsets into each logical server definition. Partitioning allows you to manually load balance within your cluster.

The Installed Packages folder lists the packages that are installed in a particular server. Even if a package is defined in a server’s repository, and all files required for components in the package are available to the server, the server does not allow creation of instances of components for packages that are not installed in the server.

The following is a partitioning example:

- Cluster “MyCluster” has three members:
  - iiop://host1:9000;iiop://host2:9000;iiop://host3:9000
- Server iiop://host1:9000 has package “P1” installed.
- Server iiop://host2:9000 has package “P2” installed.
- Server iiop://host3:9000 has packages “P1” and “P2” installed.


Note: You may want to use partitioning to separate CPU-bound components from database-bound components.

High availability

High availability provides access to your business components and Jaguar services even if a Jaguar server is unavailable.

You can use clusters to achieve high availability if:

- Each cluster has at least two member servers.
- Partitioning is used, and each partition has at least two member servers.
Automatic failover

- Each cluster has at least two name servers (which either can be members of the cluster or can be external to the cluster).

To guarantee end-to-end high availability, clients should use URLs of the form “iiop://host1:9000;iiop://host2:9000;...” when doing any of the activities listed below.

In the first three cases, the client's URL list should be the cluster's server list, or a subset of that list.

- Setting the location property for the PowerBuilder connection object.
- Setting the URL for the creation of a SessionManager::Manager object (for C++ and Java CORBA clients).
- Setting the InitialContext Provider URL property for Java clients using JNDI.
- Setting the NameServiceURL during ORB initialization. In this case, the client’s URL list should be the cluster’s name server list, or a subset of the list.

This ensures that no part of a client's initialization is limited to a single point of failure.

Note You can support non-Jaguar clients (that do not support the Jaguar-proprietary multiple URL form) by creating a Jaguar service component that, upon server start-up, writes a file containing the stringified IOR for a multiserver URL. This IOR file can then be read by any client using an HTTP connection.

Automatically failover

You can use Jaguar Manager to mark selected components to support transparent automatic failover. If a client has an object reference for a component on a server that is a member of a cluster, the client's object reference will provide transparent failover, unless all the servers in the cluster fail.

Note To avoid a single point of failure for a cluster, set the com.sybase.jaguar.cluster.startup cluster property to check_servers. See “Cluster start-up options” on page 72 for more information.
Automatic failover is not the default for Jaguar components. When a client is using a component that does not support automatic failover, and the server hosting that component fails, the client must create a new instance of that component to recover from the failure (which typically presents itself as a CORBA COMM_FAILURE system exception). However, the client does not need to create a new session, since the SessionManager::Session object supports automatic failover. The SessionManager::Session object is used implicitly by the PowerBuilder connection object and by the Jaguar COM (ActiveX) proxy.

❖ To set automatic failover for a component from Jaguar Manager:
1. Locate and highlight the component you want to set.
2. Select File | Component Properties.
3. Select the Transactions tab.
4. Select the Automatic demarcation/deactivation check box and the Autofailover check box.
5. Click OK.

Component guidelines

The following guidelines may be useful when you are writing components that support automatic failover.

The component should not retain conversational state in server memory (component instance variables), since the conversational state cannot be restored when a remote method call fails over from one server to another.

The following example shows why this would not work:

1. The client calls method A on component C on Server1. Method A retains some state in the instance in Server1's memory.

2. The client calls method B on the same component. Server1 has failed, so the client transparently fails over to Server2 and calls method B on a newly instantiated instance of component B in Server2. Since method A has not been called on this instance, it does not hold the saved state.

If you must save state between calls, consider saving it in a database. For example, in an Internet shopping application, a “shopping cart” might be represented by a database entity, and every method call on the ShoppingCart component can save the appropriate changes to the database.
In other cases, you might want to code the client to use IDL structure and sequence types to pass a list of values to a single component method, instead of passing each value in a separate call and having the component attempt to collect the list of values using conversational state. This approach also reduces network traffic, and can greatly improve response times.

Duplicate database inserts or updates can result from the use of automatic failover, as in the following example:

1. The client calls method `insertStuff` on component C on Server1.
2. The `insertStuff` method inserts a record into a database.
3. The transaction is committed.
4. The server crashes before sending the reply message over the network to the client.
5. The client transparently fails over, and calls method `insertStuff` on a new instance of component C on Server2.
6. The `insertStuff` method inserts a new (duplicate) record into the database.

   Everything works this time, but we now have a duplicate record in the database.

A simple design approach can help avoid such problems. Add a method to component C to generate a new ID for insertion: for example, `newStuffId`:

1. The client calls `newStuffId` to get a new unique ID. If you do not permit gaps in the ID numbering sequence, you cannot use this approach.
2. The client then calls `insertStuff`, passing the `StuffId` as a parameter.
3. `insertStuff` verifies that a record for that `StuffId` has already been inserted (or the database insert fails if `StuffId` is a unique key in the database).

   Although `insertStuff` has been called twice, only one database change has been made.

A component that supports automatic failover can use other components or resources that do not directly support automatic failover.
Implementing Sybase Failover for high availability systems

You can implement the Sybase Failover for High Availability Systems (failover) feature in Adaptive Server Enterprise 12.0 with Jaguar server database connectivity using Java Connection Management (JCM).

You configure JCM by enabling Java Database Connectivity (JDBC) connections to establish failover-enabled connections to Adaptive Server Enterprise 12.0. jConnect requires that the connection’s attributes be configured on an LDAP server and accessed using JNDI. See the jConnect and JNDI documentation for more details on the jConnect configuration for LDAP.

**Note** jConnect 4.2 (EBF 8719 or later) and JNDI 1.2 are required. You can download jConnect 4.2 (EBF 8719 or later) from http://www.sybase.com/products/internet/jconnect/jdbcreg42_52.html

You can implement failover using either:

- JDBC 1.0 and 2.0, or
- JDBC 2.0 extn/JTA drivers.

**JDBC 1.0 and 2.0**

To set up jConnect to access LDAP and JNDI with JDBC 1.0 and JDBC 2.0 connections, you must:

- Configure LDAP
- Configure a connection cache

**Note** In the examples in this section, you must enter the text of an indented line on the same line as the previous one.

**Configure LDAP**

To configure LDAP, run your Jaguar server with JDK 1.2 and use your LDAP software to set up an LDAP entry of the form:

```
servername=primaryserver, o=Sybase, c=US
1.3.6.1.4.1.897.4.2.5: TCP#1#hostname portnum
```
Implementing Sybase Failover for high availability systems

1.3.6.1.4.1.897.4.2.10: REQUEST_HA_SESSION=true
1.3.6.1.4.1.897.4.2.15: servername=secondaryserver, o=Sybase, c=US
    TCP#1#aeon 9898

Configure a connection cache

Use Jaguar Manager to configure the connection cache with these values for the specified properties:

- **Username**: DBuid
- **Password**: DBpasswd
- **Connection URL (servername)**:
  jdbc:sybase:jndi:ldap://<ldapserver>:<ldapport>/servername=primaryserver; o=Sybase; c=US

**Note** You must specify a semicolon (;) as the field separator in the Connection URL field.

Changing the default values or adding properties

To change the default values or specify additional properties that Jaguar uses for LDAP server configuration, you can create `<cachename>.props` files in the `$JAGUAR/Repository/ConnCache` directory as shown in the following example:

```java
chacename.props:
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
java.naming.factory.object=com.sybase.jdbc2.jdbc.SybObjectFactory
java.naming.provider.url=ldap://duplo:389/o=Sybase,c=US
```

where `duplo:389` is the port where the LDAP server is running. The values listed for the first two properties are the defaults used by Jaguar.

After creating this file, you must specify the `java.naming.provider.url`.

**JDBC 2.0 extn/JTA drivers**

You set up JDBC 2.0 extn / JTA drivers as follows:
Configure LDAP

Using your LDAP software, set up an LDAP entry of the form:

```
servicename=serverwithxa, o=Sybase, c=US
1.3.6.1.4.1.897.4.2.5: TCP#1#hostname portnum
1.3.6.1.4.1.897.4.2.10: REQUEST_HA_SESSION=true
1.3.6.1.4.1.897.4.2.18: XADataSource
1.3.6.1.4.1.897.4.2.15: servicename=secondarywithxa,
  o=Sybase, c=US
```

Configure an XA resource

Use Jaguar Manager to configure the XA resource with these values for the specified properties:

- **Username:** DBuid
- **Passwd:** DBpasswd
- **Connection URL (servicename):**
  
  LdapLookup=ldap://<ldapserver>:<ldapport>/servername=
  servicename=primarywithxa; o=Sybase; c=US

**Note** You must specify a semicolon (;) as the field separator in the Connection URL field.

**Troubleshooting the database connection**

To find out if your database connection to the LDAP server is working, ping the connection cache.

- **To ping the connection cache:**
  1. Right-click on the connection cache and select Connection Cache Properties.
2 Click Ping.

If the server does not respond or an error occurs, make sure that:

- The CLASSPATH and BOOTCLASSPATH specify the correct locations for the `ldap.jar`, `providerutil.jar`, `jconn2.jar` (the jConnect driver), and `jndi.jar` (JNDI 1.2)
- You are using JDK 1.2.
CHAPTER 5

Security Configuration

This chapter discusses the security risks involved with Internet computing and the technologies and methods used by Jaguar to protect corporate and personal data transmitted on the Internet.

To protect Internet data, Jaguar servers provide Secure Sockets Layer (SSL) features to authenticate clients and servers, and encrypt data. Establish client-side security by using the PKCS #11 module provided by either Netscape or Sybase. Jaguar provides sample certificates and a sample security applet that allow you to test your connections.

You can also establish an administration password and create roles that allow you to limit access to packages and their components.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public-key cryptography</td>
<td>96</td>
</tr>
<tr>
<td>Jaguar security management overview</td>
<td>98</td>
</tr>
<tr>
<td>Entrust integration</td>
<td>99</td>
</tr>
<tr>
<td>Using Netscape to manage certificates on the client</td>
<td>100</td>
</tr>
<tr>
<td>Security Manager tasks</td>
<td>104</td>
</tr>
<tr>
<td>Security profiles</td>
<td>121</td>
</tr>
<tr>
<td>Listeners</td>
<td>129</td>
</tr>
<tr>
<td>Jaguar quality of protection</td>
<td>133</td>
</tr>
<tr>
<td>Administration password and OS authentication</td>
<td>137</td>
</tr>
<tr>
<td>Roles</td>
<td>139</td>
</tr>
</tbody>
</table>

Related chapters

To familiarize yourself with Jaguar’s SSL features, see the SSL tutorial described in the Jaguar CTS Getting Started guide. For information about Web application security, see Chapter 25, “Configuring Web Application Security” in the Jaguar CTS Programmer’s Guide.
Public-key cryptography

To maintain secure communications between a client and host, public-key cryptography techniques are used for:

- Authentication – verifying the identity of both the client and the server; Public-key cryptography techniques use digitally signed certificates that identify network entities.
- Encryption – modifying data so that it can be read only by the party for whom it is intended. When used with a user’s private key, certificates encrypt and decrypt messages.

Unencrypted messages are known as plain text. Encoding the contents of a message is called encryption. This encrypted message is the cipher text. Decryption is the process of retrieving the plain text from the cipher text. A key is usually required to perform encryption and decryption. A CipherSuite defines the parameters and methods supported by both the client and server that perform the encryption and decryption.

Public-key encryption uses a pair of keys for encryption and decryption. One key is secret (the private key) and the other is distributed (the public key). You send your digitally signed public key (certificate) to anyone with whom you wish to communicate using encoded data.

Messages that are sent to you are encrypted with your distributed public key and decrypted by your private key, while messages sent by you are encrypted with your private key and decrypted with your distributed public key. RSA encryption is a widely used public-key encryption system.

For more information on RSA and public-key encryption, see the RSA Web site at http://www.rsa.com.

Public-key certificates

Public-key certificates provide a method to identify and authenticate clients and servers on the Internet. Public-key certificates are administered and issued by a third party known as a certification authority (CA). A subject (individual, system, or other entity on the network) uses a program to generate a key pair and submits the public key to the CA along with identifying information (such as name, organization, e-mail address, and so on). This is known as a certificate request. The CA issues a digitally signed certificate. A digital signature is a block of data that is created using a private key.
The CA ties the certificate owner to the public key within the certificate. The subject then uses the certificate, along with his private key to establish his identity. Once this is done, whomever the subject is communicating with knows that a third party has vouched for his identity.

Figure 5-1 illustrates a scenario in which a server requires client authentication:

1. A client submits a request for, and receives, a certificate from the CA.
2. An administrator installs the CA’s certificate on the server and marks it trusted. Any client certificate signed by the same CA will now be trusted and accepted by the server.
3. The client supplies its certificate and negotiates a secure connection with the server.

**Figure 5-1: Using a CA certificate for a secure connection**

SSL, HTTPS, and IIOPS

SSL provides security for network connections. Specifically, SSL uses public-key encryption to provide:
• Client and server authentication using certificates
• Encryption, which prevents third parties from understanding transmitted data
• Integrity checking, which detects whether transmitted data has been altered

Packets for other protocols can be embedded inside of SSL packets. A connection in which the application protocol is embedded inside of SSL is an SSL-tunnelled connection.

Both IIOP and HTTP can be tunnelled inside SSL, which means that these protocols take advantage of SSL security features. For example, HTTPS connections embed HTTP packets inside of SSL packets. Your Web browser creates a secure HTTP connection any time you load a page from a URL that begins with “https:”.

You can configure Jaguar to accept client connections over secure protocols IIOPS and HTTPS using:

• Security Manager to manage key pairs and certificates for Jaguar. See “Security Manager tasks” on page 104 for more information about Security Manager.
• Jaguar Manager to define security profiles that establish various levels of security on Jaguar and assign them to a listener. Profiles allow you to determine:
  • Client and server authentication requirements
  • Encryption and decryption algorithms
See “Security profiles” on page 121 and “Listeners” on page 129 for information on establishing security profiles and assigning them to Jaguar listeners.
• Jaguar to use certificates and listeners to authenticate clients, if necessary, and encrypt and decrypt data.
Entrust integration

Jaguar integrates an Entrust public-key infrastructure (PKI) that enables Jaguar servers and clients to use Entrust IDs for client/server authentication. To assign an Entrust ID (Entrust profile) to a Jaguar listener:

1. Install and use Entrust/Entelligence software to manage Entrust keys and obtain an Entrust ID. See the Entrust documentation for more information.

2. Use Jaguar Manager to configure a security profile that specifies the Entrust ID you obtained in the previous step. You can configure the security profile to accept either non-Entrust clients or only clients that supply an Entrust ID. See “Configuring security profiles” on page 125 for more information.

3. Assign the security profile to a listener. See “Configuring listeners” on page 131 for more information.

The current version of Jaguar does not use Entrust encryption operations other than SSL signing by the private key.

There are three usage scenarios involving Entrust IDs and non-Entrust certificates:

- Both client and Jaguar server use non-Entrust certificates
- Entrust client and non-Entrust Jaguar server (and vice versa)
- Both client and Jaguar server use Entrust certificates

Both client and Jaguar server use non-Entrust certificates

In this scenario, you use Jaguar’s Security Manager to access the Sybase PKCS #11 token to manage Jaguar’s keys and certificates. On the client, you use either the browser’s mechanism to manage keys and certificates for Java applets or the standalone Security Manager to access the Sybase PKCS #11 token to manage keys and certificates for C++ and Java applications.

See “Security Manager tasks” on page 104 for information about Security Manager and “Using Netscape to manage certificates on the client” on page 100 for information about Netscape certificate management. See the Jaguar CTS Programmer’s Guide for information about configuring C++ and Java applications to use certificates.
Using Netscape to manage certificates on the client

**Entrust client and non-Entrust Jaguar server (and vice versa)**

In a mixed environment of Entrust IDs and non-Entrust certificates, each side (client and server) must import the other’s CA certificate so that it will be recognized and accepted as coming from a trusted CA. For example, import the Entrust CA certificate into the non-Entrust server’s PKCS #11 token using Security Manager (the Entrust CA certificate is imbedded in the user profile’s `.key` file). Mark the CA certificate trusted.

See “Installing and exporting certificates” on page 116 for information about importing CAs and “Viewing certificate, trust, and export information” on page 119 for information about marking certificates as trusted.

You can then use the certificates and Entrust IDs as follows:

- **Client side** – client applications establish security through the ORB/global property or callback feature. See the Jaguar CTS Programmer’s Guide for more information.
- **Jaguar side** – to allow non-Entrust clients, select the allow non-Entrust client check box when you configure a security profile. See “Security profiles” on page 121 for more information.

**Both client and Jaguar server use Entrust certificates**

When both the client and Jaguar server use Entrust IDs, use Entrust to manage the IDs and use Jaguar Manager to establish a security profile that uses those IDs.

See “Configuring security profiles” on page 125 for information on configuring security profiles to use either Entrust IDs or non-Entrust certificates and enabling non-Entrust clients to connect to a listener using Entrust IDs.

**Using Netscape to manage certificates on the client**

PKCS #11 is an RSA standard that specifies an API called Cryptoki (pronounced crypto-key; short for “cryptographic token interface”), that performs cryptographic functions, such as key-pair and certificate management.
Netscape 4.0x supplies a PKCS #11 module that allows you to manage the client-side certificates. Sybase also provides a PKCS #11 module that allows you to manage your certificates. Sybase recommends that you install the Sybase PKCS #11 module into Netscape, which provides immediate access to the Jaguar sample server certificates.

**Installing Sybase PKCS #11 into Netscape 4.0x**

Start Netscape 4.0x, then:

1. Select Communicator | Security Info from the window. Or, you can click the Security icon (the padlock) in the toolbar.
2. Click on Cryptographic Modules.
3. Click Add. You see a new dialog, Create a New Security Module.
4. For Security Module Name, enter “Sybase PKCS”.
5. For Security Module File, type the full path to your libjsybcki file, then click OK. For example, on NT, enter:
   
   \i:\Program Files\Sybase\Jaguar CTS 3.0\dll\libjsybcki.dll

   On UNIX, enter:

   \/work/JagPKS/lib/libjsybcki.so

6. You should see a prompt asking for a Sybase password or PIN. Enter “sybase”. If you do not see this prompt, verify the path to the DLL/shared object.

7. After entering the password, you see Sybase PKCS listed as a security module. Click on the Sybase PKCS module, then select View/Edit. A new window, the Edit Security Module window, displays. This window contains controls for the Sybase PKCS module.

8. Click “More Info” in the new window, and verify that the state is “Ready” in the Token/Slot Information window. Click OK to close the Token/Slot Information window.

9. You can change the Sybase PKCS module password by clicking Change Password in the Edit Security Module window. Click OK to close this window.
Using Netscape to manage certificates on the client

When both Jaguar and Netscape run on the same machine, they share Sybase PKCS #11 database files. If you change the PIN, you must use the new PIN when you log in to either Jaguar or Netscape. Sybase suggests that you change your PIN through Security Manager, which automatically propagates the PIN changes to the security profiles. If you change the PIN through Netscape, you must also change the PIN in all of the security profiles; otherwise Jaguar secure listeners using those security profiles may not start the next time you restart the server. See “Changing the user PIN” on page 104 for information about changing the PIN in Security Manager.

If you modify the PIN through Security Manager, you need to restart Netscape for the changes to take effect. If you modify the PIN through Netscape, while the Jaguar server is running, shut down and restart the server. Supply the new PIN to connect to Security Manager.

Obtaining a key pair and certificate

Sybase PKCS #11 includes two sample server certificates. You can use these certificates when communicating with Jaguar from your browser. You can also request new certificates from your CA or certificate server and install them in your browser.

In general, the steps involved in obtaining a certificate from a CA are:

1. Initiate a certificate request operation by connecting to a Web server hosting the public-key infrastructure (PKI) administration HTML pages for the CA/PKI.

2. On the PKI administration Web page, complete the identity information form required by the CA.

3. Netscape generates a key pair, and stores the private key in the PKCS #11 module. The public key is digitally signed and forwarded to the CA.

   You must be logged in to the PKCS #11 module to store or access the private key.

4. The CA approves the request, generates the certificate, and makes the certificate downloadable by way of a URL.

5. The CA notifies you of certificate approval and URL location through e-mail.
6 Open the URL to get the certificate. You must log in to the PKCS #11 module using the previously established PIN. The browser automatically installs the certificate, locates the previously stored matching private key using the key ID, and sets the user-specified label.

SSL certificate information in servlets

This section describes how to include the client’s certificate information into Java servlets that are hosted by Jaguar.

If you are using JDK 1.2, you can obtain SSL certificate information about the client as follows:

```java
java.security.cert.X509Certificate peerX509;
ServletRequest request;

peerX509 = (java.security.cert.X509Certificate)
request.getAttribute("javax.servlet.request.X509Certificate");
```

Where `request` is the `ServletRequest` parameter passed in the `doXXX()` method.

See section 5.7 of the Java Servlet Specification version 2.2 for more information.

If you are not using JDK 1.2, you can obtain SSL certificate information about the client as follows:

```java
import CtsSecurity.*;

CtsSecurity.X509Certificate peerX509;

peerX509 =
```

Security Manager tasks

Security Manager allows you to manage keys and certificates used by Jaguar.

- “Security Manager management” on page 104
- “Test CA management” on page 105
- “Key management” on page 111
- “Certificate management” on page 112

Security Manager management

This section describes the tasks involved in accessing and managing the Sybase Security Manager.

You can install and use the standalone Security Manager on a client machine to manage client keys, certificates, and trust information in a local database. The client Security Manager is completely independent of Jaguar Manager and Jaguar server. Except for the login screen, the standalone Security Manager is identical to the version used to manage Jaguar server keys and certificates.

The client’s Security Manager allows C++ CORBA clients and Java applications to access Jaguar servers using SSL features over IIOPS connections. For more information, see the *Jaguar CTS Programmer's Guide*.

To start Security Manager, see “Jaguar Manager and Security Manager” in the *Jaguar CTS Getting Started* guide.

- Changing the user PIN

  The initial PIN for the PKCS #11 token is “sybase”. You can also use the same PIN to log in to Security Manager and, if installed, the Sybase PKCS #11 token in Netscape. To change to a more secure PIN:

  1. Select the Private Keys folder.
  2. Select File | Change PIN.
  3. Enter and verify the new PIN.

  Restart Netscape for the new PIN to propagate to the Sybase PKCS #11 token.

- Displaying PKCS #11 module information

  1. Select the Private Keys folder.
Chapter 5  Security Configuration

2 To view information about the Sybase PKCS #11 module, including the library version and the Cryptoki version, select File | Module Information.

To view information about the Sybase PKCS #11 token that manages your key and certificate information, including status and version information, select File | Token Information.

❖ Logging out of the PKCS #11 module
1 Select the Private Keys folder.
2 Select File | Logout.

You are still logged in to Jaguar Manager but can no longer access keys or certificates.

Test CA management

The test CA is a signing authority that signs user certificate requests. These certificates can be used by clients and Jaguar to test the security features of your applications. Certificates signed by the test CA are not intended for commercial applications. If you already have an in-house CA or other signing authority, you may not need to use the test CA.

This section describes the tasks involved in managing the test CA, including:

- “Creating a test CA” on page 105
- “Generating a user certificate signed by the test CA” on page 106
- “Processing a certificate request” on page 108
- “Exporting the test CA certificate” on page 110

Note  The test CA must exist before you can access the Process Certificate Request and Generate User Test Certificate options.

❖ Creating a test CA

To verify that the test CA is available, highlight the CA Certificates folder. You should see the Sybase Jaguar User Test CA on the right side of the window. If not, you must generate the test CA.

1 Select the CA Certificates folder.
2 Select File | Generate Test CA.
The Sybase Jaguar User Test CA displays on the right side of the window. You can now generate test certificates signed by the test CA and process certificate requests.

- **Generating a user certificate signed by the test CA**
  1. Select the CA Certificates folder.
  3. Supply the required information described in Table 5-1 on page 107. Click Back and Next to review and modify information.
  4. You can use any of the following characters for the label:
     - Letters A-Z and a-z
     - Numeric values 0-9
     - (space) ’ ( ) + , - . / : = ?
  5. Click Finish to exit the wizard and generate the certificate.
  6. Click OK in the Info dialog box. The certificate displays when you highlight the User Certificates folder.
### Table 5-1: User test certificate information

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Strength</td>
<td>Select the authentication key strength. The greater the number, the stronger the encryption. Your options are: • 512 bits • 768 bits • 1024 bits</td>
<td>For international users, key strength is 512.</td>
</tr>
<tr>
<td>Key Label</td>
<td>The name that identifies the certificate.</td>
<td>Required field. The label must be unique among all labels used for all certificates.</td>
</tr>
<tr>
<td>Validity Period</td>
<td>From the drop-down list, select the length of time that the certificate is valid.</td>
<td>When a client (or Jaguar) presents a certificate for authentication, Jaguar (or the browser) checks to see if the certificate has expired.</td>
</tr>
<tr>
<td>Cert Usage</td>
<td>Click the check box for either or both: • SSL Client • SSL Server</td>
<td>The same certificate can be used by a client and/or a Jaguar Server.</td>
</tr>
<tr>
<td>Common Name</td>
<td>Your first and last name.</td>
<td>Required field.</td>
</tr>
<tr>
<td>User ID</td>
<td>Any ID that would further identify you.</td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>The name of your company, university, or other organization.</td>
<td>Required field.</td>
</tr>
<tr>
<td>Organization Unit</td>
<td>The name of a department within your organization.</td>
<td></td>
</tr>
<tr>
<td>Locality</td>
<td>The location of your organization.</td>
<td>You must supply at least one of: • Locality</td>
</tr>
<tr>
<td>State/Province</td>
<td>State or province name.</td>
<td>• State/Province</td>
</tr>
<tr>
<td>Country</td>
<td>Your two-digit country code; for example, “US.”</td>
<td>• Country</td>
</tr>
<tr>
<td>Requester Name</td>
<td>The person requesting the certificate.</td>
<td></td>
</tr>
<tr>
<td>Server Admin</td>
<td>The name, if any, of the server administrator.</td>
<td></td>
</tr>
<tr>
<td>E-Mail</td>
<td>Your e-mail address.</td>
<td></td>
</tr>
</tbody>
</table>
Security Manager tasks

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark Private Key Exportable</td>
<td>Checked by default, this property allows you to export this certificate along with its private key.</td>
<td>See “Installing and exporting certificates” on page 116 for more information.</td>
</tr>
</tbody>
</table>

Note: If checked, you can later uncheck this property. Once unchecked, you cannot change this property. If unchecked, you cannot export this certificate and private key.

-❖- Processing a certificate request

Security Manager can process a certificate request generated from elsewhere. The test CA signs the request and generates the certificate.

1. Select the CA Certificates folder.
3. Paste the certificate request into the window as indicated. Here is an example of a base64 certificate request. You must include the entire contents, including the BEGIN and END lines:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIH4MlGjAgEAMD4xCjAIBgNVBAMTAWExCjAIBgNVBAoTAWExCjAIBgNVBAcTATE
CzAIBgNVBAgTaMNHMQswCQYDVQQGEwJlczBcMA0GCSqGSIb3DQE
BAQUAA0sAMEgC
QQC9f9A0ozflqIarPCC7eRdr3C0wrIg+3B2T+pe9sdgEjnc/bw
1GfxcZfYamWxg
G1KQycFqkdrRNP7f9gRCOd3xAgMBAAAGgADANBgkqhkiG9w0BAQQ
FAAANBAIElmCB
HbFdnJ0MtFda002f/Tn16FtGCh7Gs23pZ1WIUzD1GFowiuJY6iM
Dzd/1bJz5yYB+
Iv1M9Ath/zTF2eY=
-----END NEW CERTIFICATE REQUEST-----
```

4. Set the following certificate properties:
   - Format type – identifies the format type of the request, either “base64” or “binary.”
   - Cert Usage – depending on how you will use the certificate, select SSL Client, SSL Server, or both.
   - Validity Period – select the length of time that the certificate is valid.
5 Click Next. The certificate is generated and displays in the dialog. Here is the signed base64 certificate:

```
-----BEGIN CERTIFICATE-----
MIICYTCACqAwIBAgIBBzANBgkqhkiG9w0BAQQFADCBgjEzMDE
YXN1EphZ3VhciBvcmFkaWFpY2h5b3J5IDAQBgkqhkiG9w0BAQQF
ExdTeWJhc2UgIzAIIBhMDQwMCAwHgYDVQQLExdTeWJhc2UgSmF
ElQIjAQIIBhMDQwMCAwHgYDVQQLExdTeWJhc2UgSmFndWFyIF
ExdTEWJhc2UgIzAIIBhMDQwMCAwHgYDVQQLExdTeWJhc2UgSmF
ElQIjAQIIBhMDQwMCAwHgYDVQQLExdTeWJhc2UgSmFndWFyIF
-----END CERTIFICATE-----
```

6 Select Save to File and enter the full path name to save the generated certificate as a file. You can also select Browse to specify the location for the file.

If you want to use this certificate for authentication, you must install the certificate on the same machine that generated the certificate request, since this is where the private key is stored.

**Note** Certificates signed by the test CA are intended for testing only. In a real-life situation, the CA would verify user information to establish identity.
Security Manager tasks

- **Exporting the test CA certificate**

  You can export certificates, including the test CA certificate. Exporting the test CA certificate allows you to load it into Netscape 4.0x browsers and mark it trusted. This prevents Netscape from displaying warnings about untrusted certificate authorities when you use listeners that use certificates signed by the test CA.

  1. Select the CA Certificates folder.
  2. Highlight the Sybase Jaguar User Test CA.
  4. From the Export Certificate wizard, select the format type for the exported certificate. For the Test CA, select Binary Encode X509 Certificate. Click Next.
  5. Select Save to File and enter the full path name to a file that will contain the test CA.
     
     Do not add any extension to the file name. A .crt extension is automatically added to the exported certificate by Security Manager. Netscape 4.0x recognizes this extension as a X.509 certificate and handles it accordingly.
  6. Click Finish to export the certificate to the file you specified.

  For general information about the Export Certificate wizard and certificate types, see “Installing and exporting certificates” on page 116.

- **Loading the test CA’s certificate into Netscape 4.0x**

  You must be logged in to the Netscape token.

  1. Enter the full path of the file that contains the exported test CA’s certificate in Netscape’s URL/Netsite field.
  2. Select Open and click OK.
  3. Click Install Certificate. Netscape recognizes the .crt extension as belonging to a certificate authority and displays a series of dialog boxes asking if you want to accept the CA.

     If Netscape does not recognize the .crt file extension, perform these steps and restart Netscape before trying to load the test CA:

     1. From Netscape, select Edit | Preferences.
     2. Under Category, click Applications.
3 Under Description, scroll down and select “Internet Security Certificate.” Click Edit.

4 Verify that the Mime Type field contains:
   application/x-x509-ca-cert

5 Click OK.

**Note** If you are using UNIX, make sure the following line is in your ~/.mime.types file before you start Netscape:

   application/x-x509-ca-cert      crt cer ber der

This line ensures that Netscape recognizes the .crt file extension.

4 Follow the instructions in the dialog boxes to accept this certificate.

Netscape now allows you to connect to Jaguar ports that require authentication, and accepts the certificates signed by the test CA without displaying warnings.

---

**Key management**

This section describes the tasks involved in key management.

To view the private keys installed in the security module, select the Private Keys folder. The private keys display on the right side of the window.

Security Manager displays any private key that does not have a certificate associated with it, including private keys that have an outstanding certificate request. For example, you may generate a key pair and request a certificate from a CA at the same time. It may take several days to receive your certificate. In the meantime, the private key displays when you highlight the Private Keys folder.

Sybase recommends that you delete any private key that does not have an outstanding certificate request associated with it.

❖ **To view information about a private-key:**

1 Select the Private Keys folder.

2 Highlight the key whose information you want to view.

3 Select File | Key Information. The Key Information dialog box displays the length of the key.
Security Manager tasks

- To delete a private key:
  1. Select the Private Keys folder. The private keys display on the right side of the window.
  2. Select the key that you want to delete.
  3. Select File | Delete Key.

Certificate management

Security Manager comes with several preinstalled CA certificates. Jaguar accepts client certificates only if they have been signed by a trusted CA. You can modify the trust attribute for any of the preinstalled certificates. See “Viewing certificate, trust, and export information” on page 119 for more information.

- Generating a key pair and requesting a certificate
  You can generate a key pair and send the certificate request to a CA to be signed. Once the CA has signed and returned the request, you can install the certificate.
  1. Select the Private Keys folder.
  2. Select File | Key/Cert Wizard...
  3. Supply the required information, described in Table 5-2 on page 114. Use Back and Next to review or change any information.

  You can use any of the following characters:
  - Letters A-Z and a-z
  - Numeric values 0-9
  - (space) ’ ( ) + , . / : = ?
  4. Click Finish to exit the wizard. Security Manager generates the key pair and saves the certificate request to a file that you specify, or installs a certificate if you have pasted one into the certificate dialog box.
  5. Send your certificate request to a CA for signing. Depending on the CA, this could be through e-mail or by attaching to the CA’s URL.
  6. When you receive it, install the certificate. See “Installing and exporting certificates” on page 116.
The new private key appears on the right side of the window when you highlight the Private Keys folder. Once the certificate is received and installed, the private key is removed from the private key list.
### Table 5-2: Certificate request information

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Strength</td>
<td>Select the authentication key strength. The greater the number, the stronger the encryption. Your options are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 512 bits</td>
<td>For international users, key strength is 512.</td>
</tr>
<tr>
<td></td>
<td>• 768 bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1024 bits</td>
<td></td>
</tr>
<tr>
<td>Key Label</td>
<td>The name that identifies the private key/certificate.</td>
<td>Required field. The label must be unique among all labels used for certificates.</td>
</tr>
<tr>
<td>Common Name</td>
<td>This could be your first and last name or name of a university or Jaguar host name.</td>
<td>Required field.</td>
</tr>
<tr>
<td>User ID</td>
<td>Any User ID that would further identify you.</td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>The name of your company, university, or other organization.</td>
<td>Required field.</td>
</tr>
<tr>
<td>Organization Unit</td>
<td>The name of a department within your organization.</td>
<td></td>
</tr>
<tr>
<td>Locality</td>
<td>The location of your organization.</td>
<td>You must supply at least one of:</td>
</tr>
<tr>
<td></td>
<td>• Locality</td>
<td></td>
</tr>
<tr>
<td>State/Province</td>
<td>The name of your state or province.</td>
<td>• State/Province</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Country</td>
</tr>
<tr>
<td>Country</td>
<td>Your two-digit country code; for example, “US.”</td>
<td></td>
</tr>
<tr>
<td>Requester Name</td>
<td>The person requesting the certificate.</td>
<td></td>
</tr>
<tr>
<td>Server Admin</td>
<td>The name, if any, of the server administrator.</td>
<td></td>
</tr>
<tr>
<td>E-Mail</td>
<td>Your e-mail address.</td>
<td></td>
</tr>
<tr>
<td>Server Certificate Request</td>
<td>Displays the request information along with the generated public key.</td>
<td>Depending on the CA, you might be able to copy and paste the certificate request from this window into an e-mail and forward it for signing.</td>
</tr>
</tbody>
</table>
Certificate file extensions and types

When installing or exporting a certificate, Security Manager determines the type of certificate based on the file extension. The extensions and the type of certificates they represent are:

- `.p7c` – belongs to a PKCS #7 certificate chain.
- `.crt` – belongs to X.509 certificates, including CA certificates. In addition, Netscape certificate chains end with a `.crt` extension.
- `.p12` and `.pfx` – belong to transferred user certificates. Sybase’s PKCS #12 implementation generates PKCS #12 files with a `.p12` file extension. This extension is recognized by both Netscape and Internet Explorer. The prior PKCS #12 standard specified a `.pfx` file extension. You can install a PKCS #12 file that uses either extension into Sybase’s PKCS #11 token.
Security Manager tasks

- Binary and base64 encoding – certificates can either be encoded/decoded using a binary or base64 scheme. Base64 is based on an ASCII format and certificates of this type can be installed from a file or pasted into the appropriate window. Binary certificates, on the other hand, must be read from a file. The encoding scheme has no effect on a certificate’s file extension.

❖ Installing and exporting certificates

Security Manager allows you to export or import (install):

1. Certificates signed by the test CA.
2. Certificates signed by another CA.
3. Certificate chains – a certificate chain is a certificate that has been signed by a CA, which in turn has been signed by a CA, and so on. The certificate contains information that traces the path of the certificate back to the root CA (the original signer).
4. A signer’s (CA) certificate. You need to install a signer’s certificate and mark it as trusted so that Jaguar will accept certificates signed by that CA.
5. User certificates and their corresponding private key using the PKCS #12 standard.

PKCS #12 is an RSA standard that specifies a transfer syntax for personal identity information. Jaguar’s support of the PKCS #12 standard allows you to move user certificates and private keys between systems and programs that support the PKCS #12 standard, such as Netscape Communicator and Microsoft’s Internet Explorer.

Transferring versus importing and exporting:

Transferring user certificates and private keys allows you to use the certificate and private key in the target security environment. Exporting, installing, and marking a CA certificate trusted in the target security environment simply allows you to accept certificates that have been signed by that CA.

Sybase’s PKCS #12 implementation allows you to transfer certificates and private keys in either a domestic format (128 bit encryption) or international format (40 bit encryption). You can find more information about domestic and international support in “Security profiles” on page 121.
To install a certificate:

1. Select the folder that corresponds to the type of certificate you are installing.
2. Select File | Install Certificate.
3. Either paste the entire contents of the certificate into the box (base64 encoded certificates only), or click the Import from File box.

   If you select Import from File, the cut and paste area is dimmed. Use the browse feature to locate the certificate.
4. Click Install. If the certificate is of type .crt or .p7c, it is installed. If the file is a PKCS #12 type (has either a .p12 or .pfx extension) the PKCS #12 Certificate/Private Key window displays:

   1. Enter the password that allows access to the file. This is the password you entered when you exported the certificate and private key.
   2. To export the certificate and its private key at a later time you must check the Mark private key as exportable check box, which is, by default, already selected.
   3. Click Done.

The certificate is assigned to a folder based on its type:

- **User** – your certificates and other user certificates, including certificates signed by the test CA used to authenticate Jaguar. These are the certificates that have a matching private key stored in the PKCS #11 token.
- **CA** – certificates obtained from CAs. These identify the signers of certificates that Jaguar recognizes.
- **Trusted** – a subset of the CA certificates. These are the signers of certificates that Jaguar trusts. Jaguar accepts the certificates from clients that have been signed by trusted CAs. You must mark a CA as trusted before it appears in the Trusted folder. See “Viewing certificate, trust, and export information” on page 119 for more information.
- **Other** – certificates obtained from other users or organizations that cannot be identified as User or CA.

Once installed, you can assign a user certificate to a Jaguar security profile. For more information, see “Security profiles” on page 121.
After installing a signer’s certificate, mark it as trusted if you want to accept certificates signed by that signer. See “Viewing certificate, trust, and export information” on page 119 for more information.

❖ **To export a certificate:**

1. Select the Certificates folder that contains the certificate to be exported.
2. Highlight the certificate to be exported.
4. From the Export Certificate wizard, select the format type of the certificate to be exported.

   If you have chosen Export Certificate from the User Certificate folder, and you selected “Mark Private Key Exportable” when you generated the key pair and requested a certificate, the PKCS #12 option is available.

5. Depending on the type of certificate you select, one of two windows appears:
   - If you have selected a certificate format that is not PKCS #12, select Save to File and enter the full path name to a file that contains the certificate.
     Do not add any extension to the file name. The appropriate extension is automatically added to the exported certificate by Security Manager.
   - If you have selected PKCS #12, enter and confirm a password used to protect access to the exported certificate and its private key. When you try to install the certificate, you are prompted for this password; there are also several advanced options you can configure that affect the exported certificate. See “Advanced PKCS #12 options” on page 118. When you are finished, click Next.
     Select Save to File and enter the full path name to a file to contain the certificate.
     Do not add any extension to the file name. The appropriate extension is automatically added to the exported certificate by Security Manager.

6. Click Finish to export the certificate to the file you specified.

The advanced screen allows you to modify the PKCS #12 options listed below. The default settings are appropriate in most cases and should only be modified by experienced users:
• Include certificate trust chain – if the certificate is part of a chain, clicking this box adds information about the CAs in the certificate’s chain. See “Verifying a certificate” on page 120 for additional information about certificate chains.

• Private key encoding algorithm – the password-based algorithm used to protect the contents of the exported private key. The default algorithm is 40BitRC2, which is accepted by most browsers. If you want to export the private key using stronger or weaker encryption, select an algorithm from the drop-down list, but be sure that the target browser accepts the stronger encryption. Security Manager can export or import private keys that are shrouded with any of the listed algorithms.

• Certificate encoding algorithm – the password-based algorithm used to protect the contents of the exported user certificate. The default algorithm is 40BitRC2, which is accepted by most browsers. If you want to export the certificate using stronger or weaker encryption, select an algorithm from the drop-down list, but be sure that the target browser accepts the stronger encryption. Security Manager can export or import user certificates that are shrouded with any of the listed algorithms. See “CipherSuite terms” on page 123 for a description of the various encryption methods and terms.

Viewing certificate, trust, and export information

You can view the information about the certificates that you have installed and your own certificates, including identifying, trust, and usage information. To view certificate information:

1 Select the folder for the type of certificate you want to view:
   • User
   • CA
   • Trusted
   • Other

2 Select the certificate you want to view.

3 Select File | Certificate Info.

The Certificate Information dialog appears. Use the scroll bar to view all of the information.
The Certificate dialog includes a Trusted Certificate check box. Based on the policies of your organization, trustworthiness of the certificate signer, and other considerations, specify whether or not to mark a certificate as trusted. Only CA certificates can be marked as trusted or untrusted.

Certificates marked as trusted display when you select the Trusted folder.

For user certificates, an Exportable Private Key check box is provided. If this box is checked, you can export the certificate, along with its private key. To prevent future exports, you can uncheck the box. Once unchecked, the private key can never be exported. See “Installing and exporting certificates” on page 116 for more information.

❖ **Verifying a certificate**

Security Manager verifies the signature, expiration date, and validity of a certificate. If the certificate is part of a chain of certificates, it verifies each certificate in the chain.

A chain involves more than one certificate. Each certificate in the chain is signed by the preceding certificate. For the certificate to be verified, the entire chain must be verified. If a peer offers a certificate for authentication that belongs to a chain, at least one CA within the chain must be trusted for the certificate to be accepted.

To verify a certificate:

1. Select the folder for the type of certificate you want to verify.
2. Highlight the certificate you want to verify.

A dialog box appears that either verifies the certificate or informs you that verification was unsuccessful. Do not use certificates that fail verification.

❖ **Renaming a certificate**

Only the label of the certificate is changed. The content of the certificate remains the same.

1. Select the folder type for the certificate you want to rename.
2. Highlight the certificate to rename.
4. Enter the new name of the certificate. Click Done.
Deleting a certificate and its associated private key

Security Manager allows you to delete your own certificates and associated private keys, the test CA, and certificates that you have obtained from others.

1. Select the folder for the type of certificate you want to delete.
2. Highlight the certificate you want to delete.

Note If you delete the test CA, certificates that were signed by the test CA are no longer useful. In this case, you need to generate a new test CA and new certificates signed by the new test CA to test your security scenarios.

Security profiles

Security profiles define the security characteristics of a client-Jaguar session. You assign a security profile to a listener, which is a port that accepts client connection requests of various protocols. A Jaguar server can support multiple listeners. Clients that support the same characteristics can communicate to Jaguar via the port defined in the listener.

Each security profile has an associated security characteristic. A security characteristic is a name that has a set of CipherSuites associated with it. A security characteristic, along with the CipherSuites, defines these characteristics of a client/server connection:

- Protocol – all profiles use SSL version 3 as the underlying protocol. IIOPS and HTTPS traffic is tunneled through SSL.
- Authentication – whether or not authentication is used. Profiles can support:
  - No authentication – neither client nor server need to provide a certificate for authentication.
  - Server authentication – only the server needs to provide a certificate to be accepted or rejected by the client.
  - Client and server authentication – both the client and server supply certificates to be accepted or rejected by the other.
• Encryption strength and method – whether or not data is encrypted, and if so, the key strength and method of the encryption.

• International use – all CipherSuites are available domestically, but not all are suitable for export outside of the United States and Canada.

• Hashing method – the method used to create the message digest.

For example, the CipherSuite SSL_RSA_WITH_NULL_MD5 can be interpreted as:

SSL – the protocol used. All profiles use SSL.

RSA – the key exchange algorithm used.

NULL – no encryption.

MD5 – the hash method used to compute the message digest.

Table 5-3 and Table 5-4 clarify the relationship between CipherSuite terminology and security characteristics.
Chapter 5  Security Configuration

Table 5-3: CipherSuite terms

<table>
<thead>
<tr>
<th>Name</th>
<th>Defines</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL</td>
<td>Protocol</td>
<td>SSL (Secure Sockets Layer) protocol uses public-key encryption to establish secure Internet communications.</td>
</tr>
</tbody>
</table>
| RSA    | Key exchange algorithm | RSA and DH (Diffie-Hellman) are public-key cryptography systems, which define both authentication and encryption:  
  • RSA provides full encryption and authentication support.  
  • DH_anon provides only encryption support. |
| DH_anon|          |                                                                                                                                             |
| EXPORT | Suitable for export | Because of export regulations, some CipherSuites are not suitable for export. Only CipherSuites that contain the word EXPORT are suitable for international use. |
| NULL   | No encryption | Data is not encrypted.                                                                                                                      |

| DES    | Encryption algorithms | System: Key length:  
  DES                  56  
  3DES                168  
  DES40              40  
  RC4_40            40  
  RC4_128         128 | The greater the key length, the greater the encryption strength.                                                                           |
| 3DES   |          |                                                                                                                                             |
| DES40  |          |                                                                                                                                             |
| RC4_40 |          |                                                                                                                                             |
| RC4_128|          |                                                                                                                                             |

| EDE    | Encryption and decryption modes | CBC and EDE are modes by which DES algorithms are encrypted and decrypted.                                                                 |
| CBC    |          |                                                                                                                                             |

| SHA    | Hash function | SHA and MD5 are hash methods used to compute the message digest when generating a digital signature.                                        |
| MD5    |          |                                                                                                                                             |

Note  Browsers do not support anonymous CipherSuites.

There are four categories of security characteristics:

- **Simple** – the predefined characteristics sybpks_simple and sybpks_simple_mutual_auth offer authentication but no encryption.
- **Strong** – the predefined characteristics sybpks_strong and sybpks_strong_mutual_auth offer greater domestic encryption strength.
Security profiles

- Domestic – all characteristics are suitable for domestic use. Clients using international CipherSuites can connect to servers using domestic security characteristics.

- International – because of export regulations, only these characteristics are suitable for export:
  - sybpks_simple
  - sybpks_simple_mutual_auth
  - sybpks_intl
  - sybpks_intl_mutual_auth

Table 5-4 lists the name, the level of authentication, and the supported CipherSuites for each security characteristic.

**Table 5-4: Security characteristics**

<table>
<thead>
<tr>
<th>Name of characteristic</th>
<th>Authenticates</th>
<th>CipherSuites</th>
</tr>
</thead>
</table>
| sybpks_simple          | server       | SSL_RSA_WITH_NULL_SHA
                        |              | SSL_RSA_WITH_NULL_MD5 |
| sybpks_simple_mutual_auth | client/server | SSL_RSA_WITH_NULL_SHA
                            |              | SSL_RSA_WITH_NULL_MD5 |
| sybpks_strong          | server       | SSL_RSA_WITH_3DES_EDE_CBC_SHA
                        |              | SSL_RSA_WITH_RC4_128_SHA
                        |              | SSL_RSA_WITH_RC4_128_MD5 |
| sybpks_strong_mutual_auth | client/server | SSL_RSA_WITH_3DES_EDE_CBC_SHA
                            |              | SSL_RSA_WITH_RC4_128_SHA
                            |              | SSL_RSA_WITH_RC4_128_MD5 |
| sybpks_intl            | server       | SSL_RSA_EXPORT_WITH_RC4_40_MD5
                        |              | SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
                        |              | SSL_RSA_WITH_NULL_SHA
                        |              | SSL_RSA_WITH_NULL_MD5 |
| sybpks_intl_mutual_auth | client/server | SSL_RSA_EXPORT_WITH_RC4_40_MD5
                            |              | SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
                            |              | SSL_RSA_WITH_NULL_SHA
                            |              | SSL_RSA_WITH_NULL_MD5 |
| sybpks_domestic        | server       | SSL_RSA_WITH_3DES_EDE_CBC_SHA
                        |              | SSL_RSA_WITH_RC4_128_SHA
                        |              | SSL_RSA_WITH_RC4_128_MD5
                        |              | SSL_RSA_WITH_DES_CBC_SHA
                        |              | SSL_RSA_EXPORT_WITH_RC4_40_MD5
                        |              | SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
                        |              | SSL_RSA_WITH_NULL_SHA
                        |              | SSL_RSA_WITH_NULL_MD5 |
Chapter 5    Security Configuration

Configuring security profiles

This section describes how to create, modify, and delete a security profile. All of the configuration tasks require you to first access the Security Profiles folder. To do this, highlight the Security Profiles folder from Jaguar Manager.

See Table 5-5 on page 127 when creating, modifying, or deleting a security profile.

❖ To create a new security profile:


2. Enter the name of the new security profile. Click Create New Security Profile.

3. Complete the Security Profile sheet. Click Advanced to modify the default settings for the advanced SSL settings. Click Save. See “General, advanced, and Entrust profile properties” on page 127 for a description of the security profile properties.

If you are using an Entrust ID, select the Use Entrust check box. Click the Entrust Tab and provide the Entrust information required to access your Entrust ID.

The new security profile now appears on the right side of the window when the Security Profiles folder on the left side of the window is highlighted.

<table>
<thead>
<tr>
<th>Name of characteristic</th>
<th>Authenticates</th>
<th>CipherSuites</th>
</tr>
</thead>
<tbody>
<tr>
<td>sybpks_domestic_mutual_auth</td>
<td>client/server</td>
<td>SSL_RSA_WITH_3DES_EDE_CBC_SHA, SSL_RSA_WITH_RC4_128_SHA, SSL_RSA_WITH_RC4_128_MD5, SSL_RSA_WITH_DES_CBC_SHA, SSL_RSA_EXPORT_WITH_RC4_40_MD5, SSL_RSA_EXPORT_WITH_DES40_CBC_SHA, SSL_RSA_WITH_NULL_SHA, SSL_RSA_WITH_NULL_MD5</td>
</tr>
<tr>
<td>sybpks_domestic_anon</td>
<td>none</td>
<td>SSL_DH_anon_WITH_3DES_EDE_CBC_SHA, SSL_DH_anon_WITH_RC4_128_MD5, SSL_DH_anon_WITH_DES_CBC_SHA</td>
</tr>
</tbody>
</table>

The sybpks_domestic_anon profile is used for anonymous Diffie-Hellman communications. Neither the client nor the server is authenticated.
To modify an existing security profile:
1. Highlight the security profile you want to modify.
3. Modify the properties. Click Save when finished. See “General, advanced, and Entrust profile properties” on page 127 for a description of the profile properties.

To delete a security profile:
1. Highlight the profile entry you want to delete.
2. Select File | Delete Security Profile.
### Table 5-5: General, advanced, and Entrust profile properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name you give to the security profile.</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>A description of the security profile.</td>
<td></td>
</tr>
<tr>
<td>Use Entrust</td>
<td>Select this check box to use an Entrust ID instead of a certificate contained in the Sybase PKCS #11 token.</td>
<td>Selecting this check box prevents access to the certificates contained in the Sybase token.</td>
</tr>
<tr>
<td>Security Characteristic</td>
<td>Select a name from the drop-down list of predefined security characteristics to use for this profile.</td>
<td>See Table 5-4 on page 124 for a description of security characteristics and the CipherSuites they support.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the selected security characteristic.</td>
<td>Each security characteristic comes with a description of its features.</td>
</tr>
<tr>
<td>Sybase PKCS #11 Token Certificate Label</td>
<td>From the drop-down list, enter the certificate label you want to use for this security profile. If you have not provided the PIN for the Sybase PKCS #11 token, you will be prompted for one. This is the same PIN that you enter to access Security Manager.</td>
<td>If you are using an Entrust ID and click the Use Entrust check box, this property does not appear. See “Certificate management” on page 112 for more information on certificates.</td>
</tr>
</tbody>
</table>
### Security profiles

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL Cache Size</td>
<td>Specifies the number of entries in SSL session cache maintained by the server. The default cache size is 30.</td>
<td>Note: These are advanced SSL parameters. They should be set only by someone who is knowledgeable about SSL.</td>
</tr>
<tr>
<td>SSL Session Share</td>
<td>Specifies the number of concurrent users (sessions) that can simultaneously use the same session entry (ID) in the session cache. The default session share size is 10.</td>
<td>SSL reuses the previously negotiated security session parameters in a number of short-lived connections, which results in a relatively large performance gain over setting up completely new security sessions for each connection. When a security session is reused, clients avoid a CPU-intensive encryption of the premaster-secret using the server’s public key. Similarly, servers avoid a CPU-intensive decryption of the premaster-secret using its private key. By configuring these parameters, you can control SSL caching on the server side.</td>
</tr>
<tr>
<td>SSL Session Linger</td>
<td>Specifies the duration for which a session entry is kept in the SSL session cache after the last SSL session using this session ID was closed. The default session linger is eight hours.</td>
<td></td>
</tr>
<tr>
<td>Set Defaults</td>
<td>Select the Set Defaults check box to restore all of the advanced settings to their default levels.</td>
<td></td>
</tr>
<tr>
<td>Specify the Entrust INI File</td>
<td>Enter the complete path to the Entrust initialization file.</td>
<td>You can also use the browse feature to locate this file. For example, on Windows NT: %SystemRoot%\entrust.ini</td>
</tr>
<tr>
<td>Entrust User Profile</td>
<td>Enter the complete path to the Entrust user profile file.</td>
<td>You can also use the browse feature to locate this file. There is no default.</td>
</tr>
<tr>
<td>Entrust Password</td>
<td>The password to the Entrust login for this Entrust user profile.</td>
<td></td>
</tr>
<tr>
<td>Allow non-Entrust client</td>
<td>Click this check box to allow non-Entrust clients to connect to listeners that use an Entrust ID.</td>
<td></td>
</tr>
</tbody>
</table>
Listeners

A listener is a Jaguar port that communicates to clients using various protocols. For protocols that use SSL security features (HTTPS and IIOPS), you assign a security profile to the listener. The profile defines security characteristics of the listener. For protocols that do not use SSL (HTTP, IIOP, and TDS), no security profile is required.

This section describes the tasks required to configure listeners. You can:

- Create a new listener and assign a profile to it.
- Assign a profile to an existing listener.
- Modify listener settings for both secure (IIOPS and HTTPS) and unsecure protocols (TDS, IIOP, and HTTP).

Preconfigured listeners

Jaguar comes with preconfigured listeners for all protocols. Secure protocols are assigned a predefined security profile.

The default settings for the preconfigured listeners are described in the following table. Only secure listeners use security profiles.
Listeners

Table 5-6: Default listener settings

<table>
<thead>
<tr>
<th>Listener name</th>
<th>Port</th>
<th>Security profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td>8080</td>
<td></td>
</tr>
<tr>
<td>https1</td>
<td>8081</td>
<td>sample1</td>
</tr>
<tr>
<td>https2</td>
<td>8082</td>
<td>sample2</td>
</tr>
<tr>
<td>iiop</td>
<td>9000</td>
<td></td>
</tr>
<tr>
<td>iiops1</td>
<td>9001</td>
<td>sample1</td>
</tr>
<tr>
<td>iiops2</td>
<td>9002</td>
<td>sample2</td>
</tr>
<tr>
<td>tds</td>
<td>7878</td>
<td></td>
</tr>
<tr>
<td>OpenServer</td>
<td>7979</td>
<td></td>
</tr>
</tbody>
</table>

The default host for these listeners is “localhost.” Sybase recommends that after you install Jaguar, you log in to Jaguar Manager and change the default host setting to the actual host name or IP address of your machine. If you do not, only connection requests originating from the Jaguar host machine are accepted. This means that, until you modify your settings, Jaguar Manager must also be on the same machine as the Jaguar server. You can also modify port number settings for the preconfigured listeners. For more information, see “Modifying an existing listener” on page 131.

The OpenServer listener is intended for migrating existing Open Server applications to Jaguar. See the Jaguar CTS Programmer’s Guide for more information.

Note You must restart the Jaguar server for your changes to take effect. If you have changed the Jaguar server’s host name and port number, you must also restart Jaguar Manager and reconnect to the Jaguar server using the new host name and port number.

Listener failover

If a server cannot retrieve listener information from the repository for an IIOP listener or if an IIOP listener has not been configured, the server attempts to open a listener at this address:

IIOP: localhost, 9000

Listener start-up can fail if a port is already in use. You can verify the listener addresses in use by viewing the initial log entries in the srv.log file. If the log messages indicate a listener configuration problem, use Jaguar Manager to connect to the indicated IIOP address and reconfigure the server’s listener properties.
Configuring listeners

This section describes how to create, modify, and delete a listener. All of the configuration tasks require you to first access the Listeners folder from Jaguar Manager:

1 Double-click the Servers folder.
2 Double-click the server for which you want to create, modify, or delete a listener.
3 Click the Listeners folder on the left side of the window.

❖ Creating a new listener

1 Select File | New Listener.
2 Enter the name of the new listener, then click Create New Listener.
3 Complete the information in the Listener Info window. See Table 5-7 on page 131.

The new listener appears on the right side of the window when you highlight the Listeners folder.

❖ Modifying an existing listener

1 Highlight the listener you want to modify.
2 Select File | Listener Properties.
3 Make your modifications and click Save. Listener properties are described in Table 5-7 on page 131.

❖ Deleting a listener

1 Highlight the listener you want to delete.
2 Select File | Delete Listener Profile.

Table 5-7: Listener profile properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Select the protocol from the drop-down list: • HTTP • IIOP • TDS • HTTPS • IIOPS</td>
<td>HTTPS and IIOPS are secure protocols that provide all of the security features made available by SSL, including authentication and encryption. TDS, IIOP, and HTTP do not provide encryption. TDS and IIOP provide user name and password-based authentication.</td>
</tr>
</tbody>
</table>
### Listeners

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Comments/example</th>
</tr>
</thead>
</table>
| Host                      | The name or IP address of the Jaguar host to which the listener is being assigned. | For predefined listeners, change the initial setting from “localhost” to the actual machine name or IP address. This allows clients from other machines access to Jaguar.  
  Note Sybase recommends that you provide the IP address of the host instead of the host name. In certain cases, a client may not be able to resolve a host name. For example, the client’s DNS server or *hosts* file does not have an entry for the specified host. |
| Port                      | The port number on the host to which the listener is assigned.                | Make sure that the port is not in use by any other service.                       |
| Jaguar Security Profile   | Select one of the preconfigured security profiles from the drop-down list. This field is enabled for only the secure protocols (HTTPS or IIOPS). | You can create new security profiles that can be assigned to a listener. See “Security profiles” on page 121 for information on security profiles. |
| Enable Open Server Events | When selected, the TDS port accepts open server client connections, if not, only MASP requests are accepted. | You must use TDS as the protocol for Open Server events. |
Jaguar quality of protection

Jaguar Manager allows you to set the quality of protection (QOP) for Jaguar packages, components, and methods. QOP establishes a minimum level of encryption and authentication that a client must meet before it can access your business logic. For example, if you do not set a QOP at the package level, all clients can access the package. You can then set a QOP that restricts access to components within that package, and a different QOP that further restricts access to methods within those components.

Note This document discusses setting server-side QOP. For information about configuring client-side QOP, see the Jaguar CTS Programmer’s Guide.

Naming service support

The client’s QOP, Jaguar listener’s security profile, and the package/component/method QOP work together to establish end-to-end security. To accommodate naming services and reduce connection time, a special CORBA component tag is set in the interoperable object reference (IOR). The naming service sends only profiles with QOPs that match a client’s QOP so that the client tries to access only listeners and packages/components/methods for which the client has a compatible QOP.

See “Interoperable object references” on page 84 for information about IORs.

Usage scenarios

Figure 5-2 illustrates two clients trying to access component A. A QOP of sybpks_strong is set for the component. To access the component, the client must use a QOP that meets the minimum requirements of the component’s QOP, and communicate with a listener that also meets the minimum requirements of the component’s QOP. In Figure 5-2:

- Client 1 accesses the server at listener port 9001, but cannot access the component because the client’s QOP does not meet the minimum requirements of component A.

- Client 2 accesses the server at listener port 9002. The listener and client negotiate a CipherSuite that both support. The highest CipherSuite that both client and listener support uses 40-bit encryption and does not meet the minimum requirement of component A, since sybpks_strong supports only 128-bit encryption. Even though the client supports the minimum QOP required to communicate with component A, it is blocked because the listener does not support this minimum requirement.
Jaguar quality of protection

See Table 5-4 on page 124 and Table 5-8 on page 137 for more information about QOP compatibility.

- Neither client supports mutual authentication; consequently, neither can access the listener at port 9003.

If a client has a QOP that includes mutual authentication, it can access a package, component, or method that does not, as long as there is a listener available to authenticate the client and the client’s QOP meets the minimum level of security established at the package, component, or method. Figure 5-2 on page 134 illustrates this scenario.

**Figure 5-2: QOP usage**

![Figure 5-2: QOP usage](image)

**Figure 5-3: QOP-compatible listener**

![Figure 5-3: QOP-compatible listener](image)
Controlling access to methods

Assuming that a compatible listener is configured on the server, Figure 5-4 illustrates a situation in which the client:

- Cannot access method 1 because the client’s QOP does not match the minimum required by the method.
- Can access method 2 because sybpks_intl meets the security requirements of the method and component A, and the package has no QOP restrictions.
- Cannot access method 3 or 4 because it is blocked at the component level.

Setting a weaker QOP at the method than the component serves no purpose since the client will already be blocked at the component.

Figure 5-4: Using QOP to limit access to methods

In addition to setting a QOP that establishes minimum encryption requirements, Jaguar provides another QOP, syb_osauth, for operating system authentication. You can set two QOP settings at the package, component, or method level, as long as one of them is syb_osauth:

- If syb_osauth is requested by the client and is not present in the package, component, or method QOP, the client ORB returns COMM_FAILURE and the message “no suitable profiles found”.
If the client does not request `syb_osauth` and the component, method, or listener QOP requires OS authentication, it is considered compatible (for backward compatibility with Jaguar 3.x and 2.0 clients). In this case, the user name and password are used for OS authentication.

**Note** For `syb_osauth` to work properly, you must enable operating-system-based authorization server-wide (not at the listener level). If you do not, you cannot load packages, components, or methods that have the `syb_osauth` QOP set. See "Administration password and OS authentication" on page 137 for information about enabling authorization for your operating system.

In Figure 5-5:

- Client 1 has a compatible QOP and supplies a user name and password to access method 1. Client 1 can access method 2 without authentication.
- Client 2 has a compatible QOP and uses authentication to access method 1 but gets a COMM_FAILURE error if it tries to access method 2.

**Figure 5-5: Using syb_osauth**

- **Configuring QOP from Jaguar Manager**
  
  Highlight the package, component, or method for which you want to establish a QOP.
  
  1. Select File | Package, Component, or Method Properties.
  2. Select the All Properties tab and set:
     
     - The `com.sybase.package.qop` property for a package.
     - The `com.sybase.component.qop` property for a component.
Chapter 5  Security Configuration

• The com.sybase.method.qop property for a method.

3 If the property already exists, you can highlight it and click Modify. Otherwise, click Add.

4 Enter the appropriate property name in the Property Name field and one (or two if using syb_osauth) of the values from Table 5-8 in the Property Value field.

After configuring QOP, you must either refresh or restart the server for your changes to take effect.

Table 5-8 provides a hierarchy of QOP settings. For a given client to access your business logic:

• A QOP-compatible listener must be available on the server, and

• Either the same or weaker QOP or no QOP restrictions must be placed on the package/component/method.

Table 5-8: QOP hierarchy

<table>
<thead>
<tr>
<th>QOP hierarchy from weaker to stronger</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>sybpks_domestic_anon</td>
<td>Some QOP profiles overlap. For example, sybpks_domestic supports both 128-bit encryption and 40-bit encryption. If you use sybpks_domestic as a package QOP, a client QOP of sybpks_int1 meets the minimum requirement of 40-bit encryption. sybpks_strong supports only 128-bit encryption and is compatible with only one of the domestic or strong profiles. For a list of CipherSuites supported by each QOP profile, see Table 5-4 on page 124.</td>
</tr>
<tr>
<td>sybpks_simple</td>
<td></td>
</tr>
<tr>
<td>sybpks_simple Mutual Auth</td>
<td></td>
</tr>
<tr>
<td>sybpks_int1</td>
<td></td>
</tr>
<tr>
<td>sybpks_int1 Mutual Auth</td>
<td></td>
</tr>
<tr>
<td>sybpks_domestic</td>
<td></td>
</tr>
<tr>
<td>sybpks_domestic Mutual Auth</td>
<td></td>
</tr>
<tr>
<td>sybpks_strong</td>
<td></td>
</tr>
<tr>
<td>sybpks_strong Mutual Auth</td>
<td></td>
</tr>
</tbody>
</table>

Administration password and OS authentication

The “jagadmin” user has unlimited access to Jaguar Manager. For additional security, you can establish an administration password for the jagadmin user and enable operating system authentication. To access and configure these properties:

1 From Jaguar Manager, double-click the server you want to configure.

2 Select File | Server Properties.
3 Select the Security tab.

**Administration password** – you can establish an administrative password for the jagadmin user on each server. Only jagadmin can:

- Access the Jaguar Manager
- Set or reset the jagadmin password
- Enable and disable user authentication

To set the administration password:

a Select Set jagadmin Password.

b In the Administrator Password dialog, enter the password twice and click OK.

Administration password conventions and restrictions are the same as for user passwords for your platform.

**Enable OS Authentication** – if selected, this option maps Jaguar client users to operating system user names and passwords. You must supply a user name and password that is valid for the machine where the Jaguar server is running. For example, for UNIX, you would use network information service (NIS) passwords, and for NT, you would use your NT domain password. NT users can provide a domain name as part of their user name; for example, `\domain_name\username`.

To enable OS authentication on NT:

a From NT, start the User Manager.

b Select Policies | User Rights.

c Click on Show Advanced User Rights.

d Scroll through the rights list and select “Act as Part of the Operating System.”

e Use the Add button to add users and groups.

f Log out and back in to your NT system to enable authentication.

g From Jaguar, select the Enable OS Authentication option on the Server Properties’s Security tab.
To enable OS authentication on UNIX, select the Enable OS Authentication option on the Security tab.

**Note** The password for the jagadmin account is always defined in Jaguar Manager. Even if jagadmin is defined as an OS user name and OS authentication is enabled, the password defined in Jaguar Manager is required to log in as jagadmin.

*Enable User & Groups Validation* – if enabled, the user and group names are validated against their operating system user and group name before being added to any of the following folders:

- Authorized User
- Authorized Group
- Excluded User
- Excluded Group

To enable user and group validation, select the Enable User and Groups Validation option on the server’s Security tab.

## Roles

Jaguar’s authorization model is based on *roles*. Roles are defined in Jaguar Manager. Each role can include and exclude specific user names or digital IDs. If you use native operating system authentication, you can also include and exclude operating system group names; all users in the specified group are affected.

**Note** To include or exclude a digital ID, it must appear in Security Manager’s User Certificate folder or Other Certificate folder.

Roles are attached to Jaguar packages and components. A package or component’s role controls access as follows:

- If any roles are assigned to a package, the user must have all of these roles to use any component in the package.
- If any roles are assigned to a component, the user must have all of these roles to use the component.
• If roles are assigned to both a component and the package that contains it, the user must have all of the roles that are assigned to the package and component.

You must either refresh or restart Jaguar for any role changes to take effect.

❖ Refreshing Jaguar
1 Highlight the Roles folder.
2 Select File | Refresh.

❖ Defining a new role
1 Highlight the Roles folder.
2 Select File | New Role. Enter the required information in the subsequent dialog boxes:
   • New Role – the name of the role you are defining.
   • Description – the description, up to 255 characters, of the role.
   • Owner – the owner of the role.

❖ Deleting an existing role
1 Highlight the Roles icon. You see a list of existing roles.
2 Highlight the role you want to delete.
3 Select File | Delete Role. This option is available only to the owner of the role or the jagadmin user.
4 Click Yes to confirm deletion of the selected role.

Note Only the owner or a member of the Admin role can delete a role, except for the Admin role itself, which cannot be deleted. See “Admin role in Jaguar” on page 144 for more information.

❖ Modifying an existing role
1 Highlight the Roles icon. You see a list of existing roles.
2 Highlight the role you want to modify.
3 Select File | Role Properties.
4 Make your modifications and click OK.
Adding an existing role, or creating and adding a new role to a package, component, or method

1. Double-click the icon for the package, component, or method to expand the folders beneath it. Highlight the Role Membership folder.

2. Select File | Install Role. Then select one of the following options from the Role Wizard:
   - Install an Existing Role – a list of uninstalled roles appears in the dialog box. Highlight the role to be installed and click OK.
   - Create and Install a New Role – enter the name of the new role to be installed. Complete the role property sheet. The properties are described in “Defining a new role” on page 140.

Note A package, component, or method with no roles or role memberships defined has no access restrictions.

Assigning users and groups to roles

Each role can include and exclude specific user names and digital IDs. If you use native operation system authentication, you can also include and exclude operating system group names; all users in the specified group are affected.

To assign authorized users to a role of a component or a package:

1. Double-click the component or package to which the role belongs.
2. Double-click the Roles icon.
3. Double-click the role you want to add authorized users to.
4. Highlight the Authorized User folder.
5. Select File | Add Authorized User.
6. Enter the name of the authorized user in the dialog box, and click Add Authorized User. On NT, you can provide the name of the domain as part of the authorized user name; for example, \domain_name\user_name. The user is authenticated using the domain name controller for that domain.

The user’s name appears on the right side of the window when you highlight the Authorized Users folder.
To remove an existing authorized user, highlight the member and select File | Remove Member.

❖ To assign authorized groups to a role of a component or a package:
  1 Double-click the component or package to which the role belongs.
  2 Double-click the Roles icon.
  3 Double-click the role you want to add authorized groups to.
  4 Highlight the Authorized Group folder.
  5 Select File | Add Authorized Group.
  6 Enter the name of the authorized group in the dialog box, and click Add Authorized Group.

The group’s name appears on the right side of the window when you highlight the Authorized Groups folder.

To remove an existing authorized group, highlight the member and select File | Remove Member.

Note The users and groups of a role are mapped to operating system users and groups. To validate users and groups you must click Enable User and Group Validation from the server’s Security property sheet. You can only add validated groups to roles. When Enable User and Group Validation is disabled, package and component authorizations stop at the user level. There is no attempt to check group level authorization.

❖ To assign authorized digital IDs (certificates) to a component or a package:
  1 Double-click the component or package to which the role belongs.
  2 Double-click the Roles icon.
  3 Double-click the role you want to add authorized digital IDs to.
  4 Highlight the Authorized Digital IDs folder.
  5 Select File | Add Authorized Digital ID.
  6 A list of digital IDs appears. Double-click the name of the digital ID that you want to authorize and click Add Authorized Digital ID.
Only certificates that appear in the Security Manager’s User Certificates folder and Other Certificates folder can be authorized. This requires that you install the certificate using Security Manager. See “Installing and exporting certificates” on page 116 for more information.

The user’s name appears on the right side of the window when the Authorized Digital IDs folder is highlighted.

To remove an existing authorized digital ID, highlight the member and select File | Remove Member.

You can verify, export, or view information about an authorized digital ID by highlighting the digital ID and selecting the corresponding option from the file menu. See “Certificate management” on page 112 for more information about these options.

❖ To exclude users from a component or a package:
   1. Double-click the component or package to which the role belongs.
   2. Double-click the Roles icon.
   3. Double-click the role you want to exclude users from.
   4. Highlight the Excluded User folder.
   5. Select File | Add Excluded User.
   6. Enter the name of the excluded user in the dialog box, and click Add Excluded User. On NT, you can provide the name of the domain as part of the excluded user name; for example, `\domain_name\user_name`. The user is authenticated using the domain name controller for that domain.

The user’s name appears on the right side of the window when the Excluded Users folder is highlighted.

To remove an existing excluded user, highlight the member and select File | Remove Member.

❖ To exclude groups from a component or a package:
   1. Double-click the component or package to which the role belongs.
   2. Double-click the Roles icon.
   3. Double-click the role you want to exclude groups from.
   4. Highlight the Excluded Group folder.
   5. Select File | Add Excluded Group.
6 Enter the name of the excluded group in the dialog box, and click Add Excluded Group.

The group’s name appears on the right side of the window when you highlight the Excluded Groups folder.

To remove an existing excluded group, highlight the member and select File | Remove Member.

❖ To exclude digital IDs (certificates) from a component or a package:
1 Double-click the component or package to which the role belongs.
2 Double-click the Roles icon.
3 Double-click the role you want to exclude digital IDs from.
4 Highlight the Excluded Digital IDs folder.
5 Select File | Add Excluded Digital ID.
6 A list of digital IDs appears. Double-click the name of the digital ID that you want to exclude and click Add Excluded Digital ID.

Only certificates that appear in the Security Manager’s User Certificates folder and Other Certificates folder can be excluded. This requires you to install the certificate using Security Manager. See “Installing and exporting certificates” on page 116 for more information.

The user’s name appears on the right side of the window when the Excluded Digital IDs folder is highlighted.

To remove an existing excluded authorized digital ID, highlight the member and select File | Remove Member.

You can verify, export, or view information about an excluded digital ID by highlighting the digital ID and selecting the corresponding option from the file menu. See “Certificate management” on page 112 for more information about these options.

Admin role in Jaguar

Every Jaguar server contains an Admin package and an Admin role. You must be a member of the Admin role to run Jaguar Manager or Security Manager.

Initially, only jagadmin is a member of this role. The jagadmin user can set up additional members.

Other Admin package and Admin role characteristics are:
• When you create a new server, an Admin role and package are also created.

• You cannot modify the Admin package. The Admin package lists only Admin role members; it does not list any components.

• The Jaguar Manager enforces read-only permissions for users who are not members of the Admin role. The install, modify, and delete options are enabled for Admin role members only.
Index

A
admin mode
   cluster synchronization 77
administration password
   Jaguar server security 138
administrative password for a cluster 66
assigning users and groups to roles 141
attention event handler 11
authentication 96
authorizing
   groups 142
   users 141
automatic failover
   component 88
   component guidelines 89
   configuring 85

B
base64
   user certificate 108
binary
   user certificate 108
bind password
   cluster 68
   bindpassword 61
bulk event handler 11

certificate information
   Security Manager 119
certificate management
   generating a key pair and requesting a certificate 112
   Security Manager 112
certificate requests
   digital certificates 96
certificate usage
   SSL client 108
   SSL server 108
certificates
   CA 117
   deleting 121
   other 117
   processing a request 108
   renaming 120
   saving 109
   signed by the test CA 106
   trusted 117
   user 117
changing the Sybase PKCS #11 PIN
   Netscape or Security Manager 102
cipher text
   encrypted messages 96
CipherSuite
   and security characteristics 124
   defining encryption and decryption parameters 96
   security profile 121
   terms 123
CLASSPATH
   environment variable 24, 25
client-side security
   managing certificates 100
   PIN 101
   Sybase security module 101
cluster
   adding servers 66
   administrative password 66
   automatic failover 88

C
CA
   See certificate authority 97
certificate authority
   certificate request 96
   certificates 117
digital signature 96
   obtaining a certificate 102
Index

bind password 68
component guidelines for automatic failover 89
configuring 66
heartbeat detection 70
high availability 87
initial name context 66
logical server name 63
name server 64
name server as ordinary member 68
participating server 63
partitioning components 87
primary property 71
primary server 63
rebinding 70
synchronization 73
synchronization properties 77
synchronizing all cluster files 78
updating a non-primary server 68
URL format 68
version property 71
component
  restarting after modifying 3
configuring
  automatic failover 85
  cluster 66
  listeners 131
  message service 38
  Security Manager 104
  security profile 125
connect event handler 11
connection cache properties 22, 23
  cache-by-name 22
  connection cache sanity 22
  CTLIB 23
  description 22
  DLLI or class name 24
  JDBC 23
  name 22
  number of connections 22
  ODBC 23
  server name 23
  service name 23
connection caches
  adding 21
  creating 21
deleting 22
managing 20
modifying 21
removing 21
conventions x
CosNaming interface 51
creating
  listeners 131
  new Jaguar server 1
  security profile 125
test CA 105
  user certificate 106
cursor event handler 11

decryption
definition 96
defining
  security characteristics 124
deleting
  certificates 121
  connection caches 22
  key pairs 112
  listeners 131
  roles 140
  security profile 126
test CA 121
digital certificates 96
digital signature
digital certificates 96
disconnect event handler 11
displaying
  Security Manager PKCS #11 module information 104
dynamic event handler 11

E

enabling
  load balancing 81
encrypted messages
cipher text 96
encryption 96
Index

encrypted messages 96
security profile 122

Entrust
configuration 128
integration into Jaguar 99
usage scenarios 100

environment variables
CLASSPATH 24, 25

error event handler 11
event handlers
server 9
excluded
groups 143
users 142, 143, 144

exporting
test CA 110

F
failover
listeners 130

File Viewer 33
httprequest.log file 33
format type
user certificate 108

G
general server properties
description 4
generating
certificate request 112

H
heartbeat detection 70
high availability
cluster 87
hot standby
Jaguar server 16
HTTP
httperror.log file 33
httprequest.log file 5
httpstat.dat file 5
logging and statistics 5

HTTP properties 4
default HTML file 4
description 4
document root 4
domain name 4
HTTPS 97
ports and listeners 129

I
IIOPS 97
ports and listeners 129
importing
test CA in to Netscape 110
initial context 11, 48
initialization event handler 11
installing
certificates 116

J
jagadmin
administration password 138
Jaguar server
File Viewer 33
Runtime Monitor 34
Jaguar server properties 2
Jaguar servers
hot standby 16

K
key management
Security Manager 111

L
language event handler 11
LDAP server 61
listeners
Index

configuring 131
creating 131
default host name 130
default settings 130
deleting 131
failover 130
Jaguar ports 129
modifying 131
preconfigured 129
properties 131
load balancing
  enabling 81
  factory IOR 84
  how it works 84
partitioning 87
partitioning example 87
randomizing client requests 84
loading
test CA into Netscape 110
localhost
default listener settings 130
logging out
  Security Manager PKCS #11 token 105
M
managing
certificates 112
key pairs 111
managing client certificates
  installing the PKCS #11 module 101
  using Netscape 100
managing connection caches 20
managing XA resources 30
message event handler 11
message service
  adding access roles and thread pools 41
  adding queues, topics, selectors, and listeners 39
cms.cache 38
cms.debug 38
cms.force 38
configuring 38
configuring queue and thread pool properties 42
deleting messages, queues, topics, roles, selectors, thread pools, and listeners 44
image type 38
MessageService.props file 38
starting the graphical interface 39
varchar datatype 38
viewing messages and statistics 43
method
  restarting after modifying 3
Microsoft DTC
  transaction options 6
modifying
  listeners 131
  security profile 126
  Sybase PKCS #11 PIN 102
module
  PKCS #11 104
N
name binding 49
name server
  cluster 64
  heartbeat detection 70
  ordinary member of cluster 68
  randomizing client requests 84
  reBinding 70
naming service properties 11
naming services
  about 47
  initial context 48
  LDAP server 61
  name binding 49
  password 61
  persistent storage 51
  transient storage 51
Netscape
  loading the test CA 110
  obtaining a certificate 102
new primary
  cluster synchronization 79
new version
  cluster synchronization 79
Index

O

option event handler 11
Oracle
  creating connection caches for 23, 24, 25
OTS/XA
  creating transactions for XA resources 31
transaction options 6
overriding synchronizations 77
overview
  cluster synchronization 73

P

package
  restarting after modifying 3
  synchronization 76
partitioning
  components 87
  example 87
persistent storage 51
PIN
  changing 102
  changing in Security Manager 104
PKCS #11 module 101
PKCS #11
  installing security module in Netscape 101
  libjsybcki.dll 101
  PIN 101
  Security Manager and Netscape sharing files 102
security module 100
preconfigured listeners
  default settings 130
  security profiles 129
primary
  cluster property 71
private key 96
processing
  user certificate request 108
profile
  security 121
properties
  listeners 131
protocol
  security profile 121
public key 96
public-key cryptography 96
public-key encryption
  CA 97
  certificate request 96
  certificates 96
digital signature 96
issuing a certificate 96
key pair 96
signing authority 96

Q

quality of protection 133

R

rebinding
  name servers 70
recognizing
  .crt file extension 111
refresh
  cluster synchronization 79
renaming certificates 120
repositories
  moving information with synchronization 74
requesting
certificates 112
restart servers
  cluster synchronization 79
restarting server after modifying
  components 3
  methods 3
  packages 3
roles
  adding to a package 141
  authorizing groups 142
  authorizing users 141
defining 140
deleting 140
excluded groups 143
excluded users 142, 143, 144
Jaguar server 139
modifying 140
RPC event handler 11
Index

RSA
  public-key cryptography  96
  Web site  96
Runtime Monitor  34

S
sample certificates
  and Netscape  101
saving
  user certificate  109
secure ports
  listeners  129
secure protocols
  HTTPS  97
  IIOPS  97
security
  adding a role to a package  141
  administration password  138
  assigning users and groups to a role  141
  authenticating  96
  authorizing groups  142
  authorizing users  141
  CA  97
  certificate authority  96
  certificate requests  96
  changing the Security Manager PIN  104
  cipher text  96
  CipherSuite  96
  decryption  96
  defining a new role  140
  deleting a role  140
  digital signature  96
  displaying Security Manager module  104
  encryption  96
  excluded groups  143
  excluded users  142, 143, 144
  IIOPS  97
  installing the PKCS #11 module in Netscape  101
  issuing digital certificates  96
  logging out of Security Manager token  105
  managing client certificates  100
  modifying a role  140
  NIS password  138
  NT domain password  138
  obtaining a certificate  102
  plain text or unencrypted messages  96
  private key  96
  public key  96
  public-key certificates  96
  public-key cryptography  96
  public-key encryption  96
  roles  139
  RSA encryption  96
  sample certificates  101, 102
  secure socket layer, HTTPS  97
  Security Manager  104
  sharing PKCS #11 files  102
  signing authority  96
  SSL public-key encryption  97
  Sybase PKCS #11 PIN  101
  Sybase security module  101
  terminology  96
  user authentication  138
  security characteristic
    categories  123
    CipherSuite  124
    defining  124
    security profile  121
Security Manager
  .crt file extension  111
  base64 certificate  108
  certificate format  108
  certificate types  117
  certificate validity period  108
  changing the PIN  104
  configuring  104
  creating the test CA  105
  deleting certificates  121
  deleting keys  112
  deleting test CA  121
  exporting the test CA  110
  generating a key pair and requesting certificate  112
  generating user certificate  106
  installing certificates  116
  Jaguar user test CA  105
  logging out  105
  managing certificates  112
  managing keys  111
  managing the security module  104
PKCS #11 module information 104
processing a certificate request 108
renaming certificates 120
saving certificates 109
test CA 105
trust information 119
user certificate information 107
verifying certificate information 120
viewing certificate information 119
viewing keys 111
security profile 121
authentication 121
CipherSuite 121
CipherSuite terms 123
configuring 125
deleting 126
domestic use 122
encryption 122
international use 122
modifying 126
protocol 121
security characteristic 121, 123
SSL 121
server
adding to a cluster 66
creating a new 1
naming service 11, 47
transaction options 6
server debugging and trace properties
log file name 9
log file size 9
truncate log on start-up 9
server event handlers 11
server log
srv.log file 9
server properties
event handlers 9, 11
HTTP configuration 4
initial context 11, 48
naming service 11, 47
server resource properties 7
servlet
synchronization 76, 77
shared connection
transaction options 6
signing authority
signing digital certificates 96
test CA 105
server log 9
SSL
client and server certificates 108
security profile 121
start event handler 11
starting listeners
failover 130
stop event handler 11
Sybase security module
using Netscape libjsybcki.dll 101
synchronization
admin mode 77
all cluster files 78
component 75
moving information between repositories 74
new primary 79
new version 79
overriding 77
overview 73
package 76
properties 77
refresh 79
restart servers 79
servlet 76, 77
url format 78
version number 79
T
terminology
security 96
test CA
creating 105
deleting 121
exporting 110
generating user certificate 106
Jaguar user test CA 105
loading in to Netscape 110
processing a certificate request 108
Security Manager 105
supplying certificate information 107
Index

trace flag properties 9
transaction options
   Jaguar server 6
   Microsoft DTC 6
   OTS/XA 6
   shared connection 6
transient storage 51
trusted
   certificates 117
trusted CA
   Security Manager 119
tunneling
   HTTPS 97
   IIOPS 97
typographical conventions x

U
unencrypted messages
   plain text 96
user authentication
   Jaguar server security 138
   NIS password 138
   NT domain password 138
user certificate 117
   saving 109

V
validity period
   user certificate 108
verifying
   certificate information 120
version number
   cluster synchronization 79
version property
   cluster 71
viewing
   certificate information 119
   key pairs 111

X
XA resources
   managing 30