Oracle® Java CAPS Security Guide

Part No: 821–2746
December 2011

120126@25097

Contents

Securing Your Java CAPS Environment

The topics listed here provide information about securing a Java Composite Application Platform Suite (Java CAPS) Release 6.3 environment.

What You Need to Know

Java CAPS Security Overview

Java CAPS applications typically handle the transmission of sensitive or critical data that require a high-level of security. When developing Java CAPS applications, it is important to secure all processes in the application life cycle, including the development, testing, deployment, migration, upgrading, and patching phases.

The following topics provide an overview of security for Java CAPS:

Security Concepts

When developing integration applications, security is crucial in not only the applications you develop, but in the environments in which you develop and deploy applications as well. Many of the Java CAPS applications you create will be web services. Due to the nature of web services and the use of open access, this adds a new set of requirements to the security considerations.

Security measures can include, but are not limited to, any of the following:

Authentication and Authorization

Authentication and authorization fulfill the very basic security requirements of verifying the identify of each user (authentication) and verifying that each user has the security permissions needed to perform certain tasks and access certain information (authorization). Java CAPS employs user authentication and authorization for installing certain Repository-based projects, deploying projects, monitoring deployed projects, and using Java CAPS applications to access information. In addition to its own security features, Java CAPS uses the authentication and authorization features of GlassFish Server.

Confidentiality

Confidentiality means keeping private information private, from the code you develop to the messages processed through the applications you create. It can also mean keeping private the identities of the parties exchanging information. This is accomplished by encrypting data and by hiding the identities of each party involved in a transaction.

Non-Repudiation

When you develop applications whose mission is to exchange data between external systems, it is imperative that the content of the messages does not change and that the integrity of the content is maintained. A digital signature can validate the message, guaranteeing that a message has not been changed since it was signed and providing non-repudiation.

Secure Transport

Secure transport means ensuring that messages remain secure while they are in transit. The most widely use transport-level security protocol is Secure Sockets Layer (SSL), also known as Transport Layer Security (TLS). Java CAPS allows you to configure design-time and runtime tools to use SSL, and you can configure many individual components in Java CAPS applications to use SSL.

Physical Security

In addition to securing information and applications, the physical hardware you used should also be secured. In a production environment, make sure that the actual hardware that hosts applications and the code is stored in a secured room with no unauthorized access.

Network Security

Network security involves preventing unwanted access to your networks, as well as misuse or modification of network resources. Authentication and authorization are the first steps to securing your network, and you can limit traffic between networks by using both hardware and software to create firewalls. In addition, anti-virus software can help keep your network free of malware.

User and Role Administration

User access to information and applications needs to be managed throughout the development lifecycle. This includes tasks such as adding and removing user accounts, assigning users to roles, removing users from roles, and adding new roles. These activities control user access to tools used in development and testing, and control access to tools, applications, and information in the production environment.

Auditing and Monitoring

Once a system is in production, an audit record should be maintained of server activity and you should be able to monitor and maintain running applications. Java CAPS provides auditing through its server log, and provides monitoring and management tools with the Enterprise Manager and the JBI Monitoring and Management API.

Security in Java CAPS

Java CAPS supports standard security protocols, such as HTTP, Secure Sockets Layer (SSL), HTTP over SSL, Web Services Security (WSIT), X.509 certificates, and so on. Java CAPS can also be configured to use security realms, including the file, certificate, and LDAP realms.

Java CAPS provides user, group, and role management tools for Repository-based applications, such as the Java CAPS Suite Installer and Enterprise Manager. Java CAPS can also take advantage of security features provided by the GlassFish Server and NetBeans for secure coding and secure message transport.

At the transport level, Java CAPS can use the security features of GlassFish and of the Repository. For example, you can configure connections to the GlassFish Server, Enterprise Manager, and Suite Installer to be through SSL. In addition, transport-level security can be defined in Adapter and Binding Component properties. At the message level you can use data

encryption, web services security (WS-Security) policies, certificates, and so on. Certain Java CAPS components can be configured for additional security, such as Adapters and Binding Components.

Related Information

Additional security information for Java CAPS can be found in the following locations. These documents are referenced throughout this book.

- *Managing Java CAPS Users*

  Provides information on the security features of the Java CAPS Repository-based products.

- *Configuring Oracle Java CAPS for SSL Support*

  Provides information on configuring Repository-based products for SSL communication.

- *Using LDAP with Oracle Java CAPS*

  Provides information on using an LDAP server to maintain security information for Repository-based products.

- *Monitoring Java EE Components in Oracle Java CAPS*

  Provides information on using Enterprise Manager to monitor and manage your running Repository-based applications.

- *Administering JBI Components for Oracle Java CAPS*

  Provides general information about using the various tools available in Java CAPS to administer JBI components, including NetBeans, the GlassFish Server Admin Console, and command-line utilities.

- *Oracle Java CAPS Management and Monitoring APIs*

  Provides information on using the Management and Monitoring API to monitor alerts for JBI applications.

- Web Services Security in Java CAPS 6

  Contains information about the security features available for web services in Java CAPS.

- Chapter 9, "Configuring Security," in *Sun GlassFish Enterprise Server v2.1.1 Administration Guide*

  Describes how to configure security using the GlassFish Server Admin Console, including information on using certificates, SSL, realms, and so on.

- Chapter 10, "Configuring Message Security," in *Sun GlassFish Enterprise Server v2.1.1 Administration Guide*

  Describes how to use GlassFish tools to help secure messages processed through your Java CAPS applications that are deployed on the GlassFish Server.

- Oracle Fusion Middleware Security and Administrator's Guide for Web Services

Provides information on working with Oracle Web Services Manager, including the web services security policies supported by Java CAPS.

Note – Security information specific to Java CAPS components, such as Adapters and Binding Components, can be found in the documentation for those products.

Analyzing Your Security Requirements

Before you begin to develop Java CAPS applications and deploy them to a production environment, determine your security needs and make sure that you take the appropriate security measures. The following sections provide a starting point for analyzing your security requirements:

- "Understand Your Environment" on page 9
- "Hire Security Consultants or Use Diagnostic Software" on page 10
- "Read Security Publications" on page 10

Keep in mind the security requirements for all the different environments in which you work, including development, testing, and production.

Understand Your Environment

To better understand your security needs, ask yourself the following questions:

- Which resources am I protecting?

  Many resources in the production environment can be protected, including information in databases accessed by the application server and the availability, resources, and applications of Java CAPS, NetBeans, and the GlassFish Server.

- From whom am I protecting the resources?

  For most web sites, resources must be protected from everyone on the internet. But should the web site be protected from the employees on the intranet in your enterprise? Should your employees have access to all resources within the Java CAPS environment? Should the system administrators have access to all Java CAPS resources? Should the system administrators be able to access all data? You might consider giving access to highly confidential data or strategic resources to only a few well-trusted system administrators. It might be best to allow no system administrators access to the data or resources.

- What will happen if the protections on strategic resources fail?

  In some cases, a fault in your security scheme is easily detected and considered nothing more than an inconvenience. In other cases, a fault might cause great damage to companies or individual clients that use the web site. Understanding the security ramifications of each resource will help you protect it properly.

Hire Security Consultants or Use Diagnostic Software

Whether you deploy the Java CAPS applications on the internet or on an intranet, it is a good idea to hire an independent security expert to review your security plan and procedures, audit your installed systems, and recommend improvements. Oracle On Demand offers services and products that can help you to secure a GlassFish Server production environment. For more information, see the Oracle On Demand page.

Read Security Publications

Read about security issues:

- For the latest information about securing web servers, Oracle recommends the "Security Practices & Evaluations" information available from the CERT Coordination Center operated by Carnegie Mellon University at `http://www.cert.org/`.
- Register your Java CAPS installation with My Oracle Support. By registering, Oracle Support will notify you immediately of any security updates that are specific to your installation. You can create a My Oracle Support account by visiting `http://www.oracle.com/support/index.html`.
- For GlassFish Server and NetBeans security advisories, refer to the Critical Patch Updates and Security Alerts page at `http://www.oracle.com/technetwork/topics/security/alerts-086861.html`.

Securing Your Environment

Creating a secure production environment means that each component that the application touches is secured, including application servers, web servers, message servers, databases, external systems, and so on. The following topics provide information about securing the environment:

Secure Tokens

Java CAPS supports authentication and authorization based on secure tokens, including the following security token types:

- Username Token

- X.509 Certificate
- SAML Token
- Kerberos ticket
- Issued Token

Securing Java CAPS Repository Components

Java CAPS provides user managements tools for securing the Repository, Enterprise Manager, the Suite Installer, and Repository-based components in NetBeans.

Securing Repository Tools

Repository security manages the Java CAPS Suite Installer and Java CAPS projects, libraries, and environments in NetBeans. You can use the security features provided by Java CAPS for Repository security or you can use an LDAP server to manage security. You can also configure the Repository to use SSL for both scenarios. Default user names and passwords are provided for the Repository tools, but you should change these accounts for a more secure environment.

- For information and instructions on setting up Repository security, see "Managing Repository Users" in *Managing Java CAPS Users*.
- For information and instructions on configuring the Repository to use SSL, see "Configuring the Repository to Use SSL" in *Configuring Oracle Java CAPS for SSL Support*
- For information and instructions on using an LDAP server to handle Repository security, see "Using an LDAP Server for Repository User Management" in *Using LDAP with Oracle Java CAPS*.
- For information and instructions on using LDAP with SSL, see "Configuring the Repository for LDAP and SSL Support" in *Using LDAP with Oracle Java CAPS*.

Securing Enterprise Manager

Enterprise Manager is an administration tool that allows you to deploy, monitor, and manage Java CAPS Java EE applications. People who use Enterprise Manager can have a variety of roles, from deploying applications, installing management applications, monitoring applications, and performing management tasks against running applications. For this reason, the Enterprise Manager User Management feature provides several predefined roles that you can assign to users to grant and restrict security permissions.

You can use the security features provided by Java CAPS for Enterprise Manager security or you can use an LDAP server to manage security. You can also configure Enterprise Manager to use SSL for both scenarios. Default user accounts are created automatically when you install Java CAPS, but you should change these accounts for your production environment.

- For information about setting up Enterprise Manager Security, see "Managing Enterprise Manager Users" in *Managing Java CAPS Users*.

- For information about configuring Enterprise Manager to use the SSL protocol, see"Configuring Enterprise Manager to Use SSL" in *Configuring Oracle Java CAPS for SSL Support*.
- For information about using an LDAP server to handle Enterprise Manager security, see "Using an LDAP Server for Enterprise Manager User Management" in *Using LDAP with Oracle Java CAPS*.
- For information about using LDAP with SSL, see "Configuring Enterprise Manager for LDAP and SSL Support" in *Using LDAP with Oracle Java CAPS*.

Security in NetBeans

When developing applications, you might want to consider taking advantage of NetBeans version control plug-ins. You can easily use NetBeans with CVS, Mercurial, and Subversion, or you can integrate your own version control system with NetBeans. Incorporating version control into your development environment ensures that proprietary information in your source code remains secure by only making it available to authorized users. Java CAPS also provides a version control system for Repository-based projects.

Securing the Production Environment

Securing the production environment includes securing the hardware and non-Java CAPS software in the environment, especially the operating system. It also means securing the GlassFish Server to which Java CAPS applications are deployed.

Securing Production Computers

A Java CAPS production environment is only as secure as the computer on which it is running. It is important that you secure the physical computers, the operating systems, and all other software that is installed on the host computers. The following are recommendations for securing the computers that host Java CAPS applications in a production environment. In addition to these recommendations, check with the manufacturer of the computers and operating systems for their recommended security measures.

Note – The domain and server configuration files should be accessible only to the operating system users who configure or execute Java CAPS components and applications.

TABLE 1 Securing Java CAPS Host Computers

Security Action	Description
Physically secure the hardware.	Keep your hardware in a secured area to prevent unauthorized operating system users from tampering with the deployment computer or its network connections.

TABLE 1 Securing Java CAPS Host Computers *(Continued)*

Security Action	Description
Log out of any Java CAPS web-based administration tools before navigating to a non-secure site.	If you are logged in to a secure Java CAPS administrative tool, be sure to log out completely before browsing to an unknown or non-secure web site. These tools include Enterprise Manager, the GlassFish Admin Console, and the Java CAPS Suite Installer.
Use a file system that can prevent unauthorized access.	Make sure that the file system on each Java CAPS host can prevent unauthorized access to protected resources. For example, on a Windows computer, use only NTFS.
Set file access permissions for data stored on disk.	Set operating system file access permissions to restrict access to data stored on disk. This data includes, but is not limited to, the following: ■ Database files ■ The directory and filename location of a private keystore ■ The directory and filename location of a Root Certificate Authority (CA) keystore Operating systems provided utilities such as unmask and chmod to set the file access permissions. At a minimum, deny read and write permissions to general users. Generated files should only have write permissions for the user who generated the file (`-rw-r--r--`).
Limit the number of accounts on the host computer.	Avoid creating more user accounts than necessary on the host, and limit the file access privileges as described above. On operating systems that allow more than one system administrator user, the host should have two user accounts with system administrator privileges and one user with sufficient privileges to run the Java CAPS server. The Java CAPS user should be a restricted user and not a system administrator user. Review active accounts regularly and when personnel leave. **Caution –** Configuration data and some URL resources are stored in clear text on the file system. A sophisticated user or intruder with read access to files and directories might be able to defeat the security measures you establish with authentication and authorization methods.
Follow best practices for establishing system administrator user account names and passwords.	For additional security, do not choose obvious user names such as **`system`** or **`administrator`** for administrator accounts. Follow these guidelines when setting passwords: ■ Passwords should be difficult to guess and guarded carefully. ■ Set a policy to expire passwords periodically. ■ Do not deploy an application that can be accessed with the default user name and no password.

TABLE 1 Securing Java CAPS Host Computers *(Continued)*

Security Action	Description
Safeguard password files.	The `-passwordfile` asadmin command specifies the name of a file that contains password entries in a specific format. These password entries are stored in clear text in the password file, and rely on the file system mechanisms for protection. For additional security, create a password alias (see below).
Use a password alias.	A password alias stores a password in encrypted form in the domain keystore, providing a clear-text alias name to use instead of a password. Use the `create-passoword-alias` asadmin command to create an alias. The password for which the alias is created is stored in encrypted form. In password files and in the domain configuration file, use the following form to refer to the encrypted password: `${alias=`*alias-name*`}`
Avoid using unencrypted passwords in command lines.	You can run certain asadmin commands with the password specified in the command line. This is a security risk because they can be easily viewed on the monitor screen by others, and they may be displayed in process listings that log the execution of those commands. Take the following precautions when entering commands: ■ Enter passwords only when prompted. If you omit the password form the command line, you should be prompted for it when the command is executed. ■ Create a password file to use for running commands. ■ When the host is running on a 64–bit Red Hat Enterprise Linux 5 platform or on any HP-UX platform, do not use the shortcut link in the home directory to start the domain. The default `start_appserver_domain1` echoes the password as you type it. Instead, use the asadmin command `start-domain` *domain_name* and enter the password when prompted. ■ When building projects from a command line, specify the security credential in the `build.properties` file instead of in the command itself (see *Deploying Oracle Java CAPS Projects*). ■ When running the `stcmsctrlutil` command, enter an asterisk (*) in place of the password so the utility prompts you for the password when it runs.
Do not run the application server as root.	The application server should run only as an unprivileged user, and never as root. The directory structure in which Java CAPS is installed should be protected from access by unprivileged users.

TABLE 1 Securing Java CAPS Host Computers *(Continued)*

Security Action	Description
Do not develop applications on a production computer.	Develop Java CAPS applications on a development computer and then deploy the applications to a development computer once they have been completely tested.
Do not install development or sample software on a production computer.	Do not install development tools on production computers. This reduces the leverage intruders have should they get partial access to a production computer.
Enable security auditing.	If the operating system of the production server supports security auditing of access to files and directories, use audit logging to track any denied directory or file access violations. If you enable audit logging, ensure that sufficient disk space is available for the audit log.
Consider using additional software to secure your operating system.	Most operating systems can run additional software to secure a production environment. For example, an Intrusion Detection System (IDS) can detect attempts to modify the production environment. Refer to your operating system vendor for information about available software.
Apply operating system security patches.	Refer to your operating system vendor for a list of security-related patches.

Securing the GlassFish Server in Production

The following table lists measures you can take to secure the GlassFish Server that is used in the production environment. For additional information about the security features of GlassFish Server, see Chapter 9 "Configuring Security" in the *GlassFish Enterprise Server v2.1.1 Administration Guide*.

TABLE 2 Securing the GlassFish Server

Header	Header
Protect the GlassFish Server password file.	If you create a domain with the `-savelogin` option, the administration user name and password are saved in the `.asadminpass` file in the user's home directory. Make sure that this file remains protected. Information in this file is used by asadmin commands to manage the domain. The same thing applies to any files you use that might include passwords, such as `build.properties` files or silent installation properties files.

TABLE 2 Securing the GlassFish Server *(Continued)*

Header	Header
Follow best practices for establishing system administrator user account names and passwords.	The Java CAPS Installer expects complex passwords for the GlassFish server with the following characteristics: ■ Contain least eight characters long. ■ Contain at least one numeric character. ■ Contain at least one uppercase character. ■ Contain at least one lowercase character. Note that the installer does not enforce these policies.
Use SSL, but do not use the self-signed certificates in a production environment.	To prevent sensitive data from being compromised, secure data transfers by using HTTPS. By default, GlassFish Server uses self-signed certificates. The self-signed certificates might not be trusted by clients by default because a certificate authority does not vouch for the authenticity of the certificate. You can instead use your own certificates as described in "Generating a KeyStore and TrustStore" in *Configuring Oracle Java CAPS for SSL Support*.
Restrict the size and the time limit of requests on external channels to prevent Denial of Service attacks.	The default setting for maximum post size is 2097152 bytes and 900 seconds for the request timeout.
Enable authentication and authorization auditing.	Auditing is the process of recording key security events in your Java CAPS environment. You can use the audit trail of the GlassFish Server to develop an audit trail of all authentication and authorization decisions. To enable audit logging, do the following: 1. On the GlassFish Admin Console, navigate to Configuration > Security. Select the Audit Logging Enabled checkbox. 2. Set the `auditOn` property for the active audit module to true.
Set logging for security messages.	Consider setting the log levels for the `javax.enterprise.system.core.security` module to log more security information. Be aware that setting finer logging levels my produce a large log file.
Ensure that you have correctly assigned users to the correct groups.	Make sure only active users have accounts for the GlassFish Server and that they are assigned to the correct groups. In particular, be sure that users assigned to the asadmin group need to be members of that group.
Create no fewer than two user accounts in the asadmin group.	The admin user is created when you install Java CAPS. For production environments, create at least one other account in the asadmin group in case one account password is compromised.

Securing Oracle Java CAPS JMS IQ Manager

You can define security for the Oracle Java CAPS JMS IQ Manager using GlassFish Server security features or through an LDAP server. You can also configure the JMS IQ Manager to use SSL. The following topics provide information on setting up security for the JMS IQ Manager:

- "Managing Oracle Java CAPS JMS IQ Manager Users" in *Managing Java CAPS Users*
- "Configuring a Oracle Java CAPS JMS IQ Manager to Use SSL" in *Configuring Oracle Java CAPS for SSL Support*
- "Using an LDAP Server for Oracle Java CAPS JMS IQ Manager User Management" in *Using LDAP with Oracle Java CAPS*

Developing Secure Applications

Developing secure Java CAPS applications is critical for many reasons. The applications you develop are often used to transmit sensitive, propriety, personal, or critical data. This data must be sent in a secure manner with its integrity guaranteed. Certain applications must follow strict regulations, such as HIPAA, the Sarbanes-Oxley Act, and so on. In addition, messages may be exchanged between many groups, including trading partners, customers, and vendors. Each organization's governance policies and security requirements must be met.

Developing secure application generally includes, but is not limited to, the following:

- Authentication
- Authorization
- Confidentiality
- Non-repudiation
- Secure transport
- User administration and role management
- Auditing and control

These security measures are all described under "Security Concepts" on page 5.

Java CAPS provides multiple implementation alternatives to support secure services and processes. Many Adapters and Binding Components support a variety of secure transports. For example, the HTTP Adapter and Binding Component both support HTTP Basic Authentication and HTTP over SSL. Many Java CAPS components can be configured to work with an LDAP server.

Additional security can come from other Oracle products, including Access Manager and Oracle Web Services Manager (OWSM). For example, Access Manager can be used to store user information and to generate and validate SAML assertions. The keystore in the application server can be used to maintain digital certificates. For information about message-level security in GlassFish Server, see Chapter 10 "Configuring Message Security" in the *GlassFish Enterprise Server v2.1.1 Administration Guide*.

When developing applications in Java CAPS, keep the following security considerations in mind.

- Security can be handled at multiple levels, so you need to determine which deployment model best secures your application. You can set security policies at the message level, at the application level, and at the application server level. You can use a combination of these to set security for your applications.
- When possible configure your applications to use SSL.
- Examine your applications for any security vulnerabilities. Organizations such as the Open Web Application Security Project have identified common issues. See the OWASP Top Ten Project for additional information.
- If your application contains untrusted code, you might want to enable the Java security manager.

Component Security Features

Several Java CAPS components provide support for transmitting data using secure protocols, and login credentials must often be configured in their properties in order to connect to external systems. For most Adapters, you can configure security in either the Connectivity Map or Environment properties. For most Binding Components, security is defined in the extensibility elements. For more information, refer to the documentation for the Adapter or Binding Component you want to use.

Below are a few links that provide examples of Java CAPS component properties for configuring security.

- LDAP Adapter:
  - "LDAP Adapter Connectivity Map Properties" in *Configuring Project Components for Oracle Java CAPS Communication Adapters*
  - "LDAP Adapter Properties" in *Configuring Environment Components for Oracle Java CAPS Communications Adapters*
- HTTP Adapter:
  - "HTTPS Adapter Environment Properties" in *Configuring Environment Components for Oracle Java CAPS Communications Adapters*
- Batch Adapter:
  - *Oracle Java CAPS Adapter for Batch User's Guide*

    The Batch Adapter supports multiple types of security protocols, including FTP over SSL, SFTP, SCP, and so on.
- SAP Adapter:
  - *Configuring Secure Network Communications for SAP*
- HTTP Binding Component:

    - "Using the Tango Web Service Features with the HTTP Binding Component" in *Oracle Java CAPS HTTP Binding Component User's Guide*
    - "HTTP Binding Component Security" in *Oracle Java CAPS HTTP Binding Component User's Guide*
    - "Configuring the Tango Web Services Attributes exposed by the HTTP Binding Component" in *Oracle Java CAPS HTTP Binding Component User's Guide*
- LDAP Binding Component:
    - "Security for LDAP Transactions" in *Oracle Java CAPS LDAP Binding Component User's Guide*
    - "Service Level WSDL Elements" in *Oracle Java CAPS LDAP Binding Component User's Guide*
    - "Service Level WSDL Elements" in *Oracle Java CAPS LDAP Binding Component User's Guide*

The following topics provide additional information about configuring SSL for specific components:

- "Using SSL With the WebSphere MQ Adapter" in *Configuring Oracle Java CAPS for SSL Support*
- "SSL and Adapters" in *Configuring Oracle Java CAPS for SSL Support*
- "Using the OpenSSL Utility for the LDAP and HTTPS Adapters" in *Configuring Oracle Java CAPS for SSL Support*
- "LDAP Adapter Connectivity Map Properties" in *Configuring Project Components for Oracle Java CAPS Communication Adapters*

Web Services Security Standards in Java CAPS

For web services, security occurs at the binding level for the binding component that exposes a business process to external clients. Frequently, this is the SOAP binding, but most binding components define some level of security. Enforcement of security policies does not generally occur in the business process itself.

Java CAPS supports the following message-level web service specification:

- WS-Security
- WS-SecurityPolicy
- WS-Trust
- WS-SecureConveration

For a discussion of web services security in Java CAPS, see the white paper Web Services Security in Java CAPS 6.

Where WS-Security is supported, Java CAPS also supports the following Oracle Web Services Manager (OWSM) policies:

- wss11_username_token_with_message_protection_service_policy
- wss11_saml_token_with_message_protection_service_policy

OWSM allows you to centrally define policies that govern web services operations (such as access policies, logging policies, and load balancing), and then wrap these policies around web services without needing to modify those services. For more information about these policies, see the following documents:

- Oracle Fusion Middleware Security and Administrator's Guide for Web Services
- Interoperability with Oracle GlassFish Enterprise Server Release 3.0.1 in *Oracle Fusion Middleware Interoperability Guide for Oracle Web Services Manager*

You can also find sample projects implementing these policies on the Java CAPS sample site at `http://java.net/projects/javacaps-samples/pages/Home`.

Auditing and Monitoring

The final aspect of security in a production environment is to monitor messages being processed through the applications and to monitor and manage the running applications, making sure that all components are up and running successfully. Java CAPS provides a monitoring application, Enterprise Manager, to monitor and manage running Java EE applications. For JBI components, you can use the Management and Monitoring API provided with Java CAPS.

The following topics provide information about monitoring Java CAPS applications:

- For information about monitoring and managing Java EE applications in Enterprise Manager, see *Monitoring Java EE Components in Oracle Java CAPS*
- For information about monitoring BPM business processes, see *Monitoring Oracle Java CAPS Business Processes*
- For information about using the Management and Monitoring API for JBI components, see *Oracle Java CAPS Management and Monitoring APIs*
- For information about managing Java CAPS JBI components, see *Administering JBI Components for Oracle Java CAPS*.

Information about the alert codes generated by Java CAPS components is provided in these topics:

- *Alert Codes for Oracle Java CAPS Adapters*
- *Alert Codes and Error Messages for Oracle Java CAPS Master Indexes (Repository)*
- *Alert Codes and Error Messages for Oracle Java CAPS JBI Components*