WebSphere® Process Server for z/OS

IBM

Version 6.0.1

Securing your Applications and their Environment

Note
Before using this information, be sure to read the general information in "Notices" on page 29.

23 June 2006

This edition applies to version 6, release 0, modification 1 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Securing applications and their environment

The security of the WebSphere Process Server environment and your applications is very important.

These documents are supplemental to the core security documentation located in the WebSphere Application Server for z/OS Information Center.

WebSphere Process Server documentation (in PDF format)

Security overview

The security of your data and processes is critical. WebSphere Process Server security is based on the WebSphere Application Server version 6.0 security. Refer to the WebSphere Application Server for z/OS Information Center for detailed information about security.

Security tasks can be broadly divided into those concerning the administration of security in the WebSphere Process Server environment and those that are related to the applications running in WebSphere Process Server. The security of the server environment is central to the security of applications, and therefore the two sides should not be thought of in isolation.

Securing the environment involves enabling global security, creating profiles with security, and restricting access to critical functions to selected users.

There are several aspects to securing an application. First is authentication; a user or a process that invokes an application must be authenticated. Second is access control; does the authenticated user have permission to perform the operation? The third aspect is that of integrity and privacy of the data that is accessed by an application. The last element is the concept of identity propagation with single sign on, which permits a user to provide authentication data once and then passes this authentication information to downstream components.

The remainder of this section details the security considerations at various stages of operation of the WebSphere Process Server.

Securing the WebSphere Process Server environment

Security in your WebSphere Process Server environment is controlled from the administrative console. A user with sufficient privileges can turn on or off all application security from the administrative console. It is therefore critical that you secure the environment before deploying secured applications.

You should install WebSphere Process Server and verify the installation before commencing these tasks.

Your WebSphere Process Server environment is defined within a profile. Open the administrative console for the profile that you want to secure. Log in to the console using any user identity; until the profile is secure, any user name will be accepted.

1. Turn on global security.
2. Assign users to appropriate administrative security roles.

The next time you log in to the administrative console you must provide a valid user name and password.

Each profile that you create must be secured in this way.

Installing WebSphere Process Server: security considerations

Complete these tasks to implement security before, during, and after installing WebSphere Process Server.

These tasks should be undertaken when you are installing WebSphere Process Server.

1. Secure your environment before installation.

   The commands required to install WebSphere Process Server with proper security will vary with operating system. For detailed information about steps to take before installing, see the topic **Securing your environment before installation** in the WebSphere Application Server Information Center.
2. Prepare the operating system for installation of WebSphere Process Server.

   This step includes information about how to prepare the different operating systems for installation of WebSphere Process Server. For detailed information about preparing your operating system for installation, see the topic **Preparing the operating system for product installation** in the WebSphere Application Server Information Center.
3. Secure your environment after installation.

   This task provides information about how to protect password information after you install WebSphere Process Server. For detailed information about securing your environment after installing, see the topic **Securing your environment after installation** in the WebSphere Application Server Information Center.

When you have completed the installation, security can be administered from the administrative console.

Related information

Preparing for security at installation time

Preparing the base operating system

Securing your environment after installation

Authentication information provided at install time

When you configure WebSphere Process Server you augment the default profile. Part of this profile augmentation process involves providing a user name and password at various portions of the response file. The user names and passwords that you provide must correspond to an identity in the user registry chosen for this profile. The user names and passwords you supply are required when you enable global security.

Several components of WebSphere Process Server utilize authentication aliases. These aliases are used to authenticate the runtime component for access to databases and messaging engines. The profile augmentation process collects a valid user name and password that is used to create these aliases.

If you do not provide this information , you must use the administrative console to enter a valid user name and password for each alias. Failure to provide valid user names and passwords will cause errors when global security is turned on.

Augmenting WebSphere Process Server profiles with security:

You can take steps to secure your environment when you augment the WebSphere Application Server for z/OS default profile with WebSphere Process Server security profile data. Alternatively you can provide the same information on the administrative console after you augment the profile.

When you configure WebSphere Process Server there are several response file properties representing components, where you can enter user names and passwords for security purposes. The three components of WebSphere Process Server that permit you to enter these user names and passwords are the service component architecture (SCA), Business Process Choreographer, and the Common Event Infrastructure (CEI).

These user names and passwords are used to create authentication aliases and are required when you enable security. If you do not enter the user names and passwords when you configure WebSphere Process Server, you can provide the same information using the administrative console, after you have configured the WebSphere Process Server.

1. In the Service Component Architecture portion of the response file, provide an identity to be used to connect components to the Service Integration Bus in a secured mode.
   a. Ensure the Service Component Architecture property value is set to true:**configureScaSecurity=true**.
   b. Enter a valid user name and password as values in the appropriate property fields.
2. On the Common Event Infrastructure portion of the response file, provide an identity to be used to authenticate with WebSphere Messaging queue manager.

   Enter a valid user name and password in the appropriate fields.
3. On the Business Process Choreographer portion of the response file, provide an identity for the sample Business Process Choreographer configuration to connect to the Service Integration Bus in a secured mode.

More information about managing authentication aliases is provided in subsequent topics.

Setting up WebSphere Process Server security

The first step to take to secure your WebSphere Process Server environment is to supply authentication information augmenting server profiles.

It is assumed that you have successfully installed WebSphere Process Server before commencing this task.

A detailed description of profile augmentation is provided in Installing WebSphere Process Server documentation.

1. As part of the product configuration process you can enter a user name and password in the response file associated with the configuration that you are creating. These user names must be in the user registry that you intend to use for authenticating users. By default this would be the local operating system user registry but you can use the Lightweight Directory Access Protocol (LDAP) if you prefer. The users associated with these components will have some administrative privileges to these components and therefore the user name provided should be someone in authority.

2. Complete profile augmentation.

   Complete response file edits and run the product configuration script.
3. Start the server.

   Start the server by running the **startServer** command from the `<install_root>/bin` directory. The command syntax is as follows:

   ```
   startServer <server>
   ```

   where <server> is the name of the application server that you are starting.
4. Enable global security.

   See See Enabling global security for more information. for more information about enabling global security.
5. Stop and restart the server.

   The server restarts with global security enabled.

Enabling global security

The first step in securing your WebSphere Process Server environment and your applications is to enable global security.

Install WebSphere Process Server and verify the installation before commencing these tasks.

Open the administrative console for the profile that you want to secure. Log in to the console using any user identity; until the profile is secure, any user name will be accepted.

1. Open the global security panel in the administrative console.

   Expand **Security** and click **Global security**.
2. Enable global security.

   Select the **Enable global security** check box.
3. **Optional:** Enforce Java 2 security, if required.

   Select the **Enforce Java 2 security** check box to enforce Java 2 security.

   When you enable Java 2 security, an application that requires more Java 2 security permissions than are granted in the default policy might fail to run properly until the required permissions are granted in either the app.policy file or the was.policy file of the application. AccessControl exceptions are generated by applications that do not have all the required permissions. For more information about Java 2 security see the topic on Configuring Java 2 security policy files in the WebSphere Application Server Information Center.
4. Apply these changes.

   Click the **Apply** button at the bottom of the panel.
5. Save the changes to the local configuration.

   Click **Save** in the message pane.
6. If necessary stop and restart the server.

   If the server needs to be restarted, a message will appear in the administrative console to this effect.

You must turn on global security for each profile that you create.

Related information

Configuring Java 2 security policy files

Setting up security for a stand-alone WebSphere Process Server

Increase the security of a stand-alone installation of WebSphere Process Server by taking the following steps.

1. Start WebSphere Process Server.
2. Launch the administrative console. If global security is not yet enabled you can use any user name to access the administrative console.
3. Enable global security.

   Expand **Security**, click **Global security**, and select the **Enable global security** check box.

   Note: Selecting the **Enable global security** check box results in the **Enforce Java 2 security** check box being checked.
4. **Optional:** Enable Java 2 security permission checking, if required.

   Select the **Enforce Java 2 security** check box to enforce Java 2 security.

   With Java 2 security enabled, an application that requires more Java 2 security permissions than are granted in the default policy might fail to run properly until the required permissions are granted in either the app.policy file or the was.policy file of the application. AccessControl exceptions are generated by applications that do not have all the required permissions. For more information about Java 2 security, see the topic on Configuring Java 2 security policy files in the WebSphere Application Server for z/OS Information Center.
5. Set the authentication mechanism to Lightweight Third Party Authentication (LTPA).

   Click **Lightweight Third Party Authentication (LTPA)** on the **Active authentication mechanism** list. LTPA is the only authentication mechanism supported by WebSphere Process Server. For more details on configuring LTPA as the authentication mechanism, see the Configuring single signon topic in the WebSphere Application Server for z/OS Information Center.
6. Enter a password to be used for LTPA key storage.

   Expand **Authentication mechanisms**, and select **LTPA**. In the **Password** field, enter a password and type the same password in the **Confirm password** field. This password is used for the LTPA key storage. Confirm your changes by clicking **Apply**.
7. Provide necessary parameters for the user registry.

   The following table describes the actions that you must take to provide required security information for the selected user registry.

Table 1. Choices of user registry and actions required to provide security information for that user registry.

User registry	Action
Operating System	Under **User registries**, choose **Local OS**. On the **Local OS user registry** page, provide a user name and password. **Note:** This user name is used as the identity of the server. The user is automatically added to the **Administrator** role.
Lightweight Directory Access Protocol (LDAP)	See Configuring Lightweight Directory Access Protocol (LDAP) as the user registry for information about configuring LDAP as your user registry.

8. Save your changes

Click **OK**.

9. Restart the WebSphere Process Server.

Related information

Configuring Java 2 security policy files

Setting up security for a Network Deployment environment

Increasing the security of your Network Deployment environment requires steps in addition to those required for a stand-alone version of WebSphere Process Server.

You must be running the administrative console on the server which is acting as the deployment manager.

Take the following steps to set up security in a Network Deployment environment.

1. Enable global security.

   On the administrative console, expand **Security**, click **Global security**, and select the **Enable global security** check box.

2. **Optional:** Enable Java 2 security permission checking, if required.

   Select the **Enforce Java 2 security** check box to enforce Java 2 security.

   With Java 2 security enabled, an application that requires more Java 2 security permissions than are granted in the default policy might fail to run properly until the required permissions are granted in either the app.policy file or the was.policy file of the application. AccessControl exceptions are generated by applications that do not have all the required permissions. For more information about Java 2 security, see the topic on Configuring Java 2 security policy files in the WebSphere Application Server for z/OS Information Center.

3. Set the Lightweight Third Party Authentication (LTPA) as the authentication mechanism.

   Click **Lightweight Third Party Authentication (LTPA)** on the **Active authentication mechanism** list. LTPA is the only authentication mechanism that is supported in WebSphere Process Server.

4. Enter a password to be used for LTPA key storage.

   Expand **Authentication mechanisms** and select **LTPA**. In the **Password** field, enter a password and type the same password in the **Confirm password** field. This password is used for the LTPA key storage. Confirm your changes by clicking **Apply**.

5. Configure LDAP as the user registry. You must use LDAP as the user registry in a network deployment environment. See *Configuring Lightweight Directory Access Protocol (LDAP) as the user registry* for details.

6. Ensure that the security information is propagated to the nodes of the cell.

   Select the **Synchronize with Nodes** check box.

7. Save your changes

   Click **OK**.

8. Restart the deployment manager, the nodes, and the servers.

Related information

Configuring Java 2 security policy files

Choosing a user registry

The user names and passwords of registered users are stored in a user registry. You can use either the user registry of the local operating system (this is the default) or the Lightweight Directory Access Protocol (LDAP).

The user registry is the user and groups account repository that the authentication mechanism consults when performing authentication. Choose a user registry on the administrative console.

Note: In a network deployment environment you can use either LDAP or your local operating system as your user registry.

1. Navigate to the global security panel in the administrative console. Expand **Security** and click on **Global security**.
2. Select the user registry you want to use.

   The following table describes the choices of user registry and the actions required to select and configure a user registry.

User registry	Action
Operating System	The default user registry. Under **User registries** click**Local OS**. On the Local OS user registry page provide a user name and password. This user name is used as the identity of the server. The user is automatically added to the **Administrator** role. **Note:** Do not use the local operating system as the user registry in a network deployment environment.
Lightweight Directory Access Protocol (LDAP)	Follow the instructions in *Configuring Lightweight Directory Access Protocol (LDAP) as the user registry* to configure LDAP as your user registry.

Configuring Lightweight Directory Access Protocol (LDAP) as the user registry

By default, the user registry is the local operating system registry. If you prefer, use an external Lightweight Directory Access Protocol (LDAP) as the user registry. In a network deployment environment you must use LDAP.

This task assumes that you have global security turned on.

1. Start WebSphere Process Server.
2. Launch the administrative console.
3. Open the LDAP User Registry configuration page.

   Expand **Security**, click **Global security**, and click **LDAP** under the **User Registries** heading.
4. Set the user name and password used to run WebSphere Process Server for security purposes.

   In the **Server user ID** field type the user name, and in the **Server user password** field, enter the corresponding password. Although this ID is not the LDAP administrator user ID, however, the entry must exist in the LDAP.
5. Choose the type of LDAP you are using.

   From the **Type** list choose the specific LDAP that you want to use as your user registry.
6. Enter the name of the computer where the LDAP resides.

   In the **Host** field, enter the name of the server where the LDAP resides.
7. Enter the port number on which the LDAP listens.

   In the **Port** field, enter the port number on which the LDAP server is listening.

8. Enter the **Base Distinguished Name**.

   This value specifies the base distinguished name of the directory service, indicating the starting point for LDAP searches of the directory service.

   For authorization purposes, this field is case sensitive. This specification implies that if a token is received (for example, from another cell or Domino server) the base distinguished name (DN) in the server must match the base DN from the other cell or Domino server exactly. If case sensitivity is not a consideration for authorization, enable the **Ignore case** field. This field is required for all LDAP directories except for the Domino Directory, where this field is optional.
9. Leave the remaining parameters with the default values and confirm your changes.

   Click **OK**.

Starting and stopping the server

When global security is enabled, to shut down the server you must provide the appropriate user name and password. The server will start without authentication, but that authentication is required to access the administrative console.

Global security must be enabled.

1. Start the server. At the command prompt in the *install_dir*/bin directory, type the following text from a command line: startServer.sh *servername*.

   Note: You are not required to provide a user name and password to start the server. However, you will need to authenticate yourself if you try to launch the administrative console or perform any other administrative task.

   The server starts or an en error message is returned.
2. Stop the server. At the command prompt in the *install_dir*/bin directory, type the following text: stopServer.sh *servername*-username *username* -password *password*

   Note: You are required to provide a user name and password to stop the server.

   If the user name and password you provide are members of the operator or administrator roles, the server will stop.
3. Check that the server stopped successfully

   The outcome of your request is displayed in the command window from which the request was made.

Administrative security roles

Four administrative security roles are provided as part of the WebSphere Process Server installation.

There are four roles provided as part of the administrative console. These roles grant permission to ranges of functionality on the administrative console. When global security is enabled, a user must be mapped to one of these four roles in order to access the administrative console.

The first user to log in to the server after installation is added to the administrator role.

Table 2. Administrative security roles

Administrative security role	Description
Monitor	A member of the monitor role can view the WebSphere Process Server configuration and the current state of the server.
Configurator	A member of the configurator role can edit the WebSphere Process Server configuration.
Operator	A member of the operator role has monitor privileges, plus the ability to modify the runtime state, i.e., start and stop the server.
Administrator	The administrator role is a combination of configurator and operator roles plus additional privileges granted solely to the administrator role. Examples include: • Modifying the server user ID and password • Mapping users and groups to the administrator role Also has permission required to access sensitive information such as: • LTPA password • keys .

The server ID that is specified when you enable global security is automatically mapped to the administrator role. Users or groups can be added to and removed from the administrative roles at any time through the WebSphere Process Server administrative console. However, a server restart is required for the changes to take effect. A best practice is to map a group or groups, rather than specific users, to administrative roles because it is more flexible and easier to administer. By mapping a group to an administrative role, adding or removing users to or from the group occurs outside of WebSphere Process Server and does not require a server restart for the change to take effect.

In addition to mapping users or groups, a special-subject can also be mapped to the administrative roles. A special-subject is a generalization of a particular class of users. The AllAuthenticated special-subject means that the access check of the administrative role ensures that the user making the request is at least authenticated. The Everyone special-subject means that anyone, authenticated or not, can perform the action, as if security was not enabled.

Elements of application security

Applications that run in WebSphere Process Server are secured by authentication and by access control. In addition the data that is transferred during the invocation of an application is kept secure by various mechanisms; these mechanisms ensure that the data cannot be read or altered in transit. The final element of security is the propagation of security information through various systems, in order that the user need not repeatedly enter a user name and password.

It is possible to divide security in WebSphere Process Server into three broad groupings:

- Application security

- Data integrity and privacy
- Identity propagation

Application security

The security of your WebSphere Process Server applications is maintained in two ways:

- **Authentication** A user who wants to use an application must provide a user name and password from the user registry.
- **Access control** A user must have permission to invoke the application. Roles are associated with invocation of the application. An authenticated user must be part of the appropriate role, otherwise the application will not run.

Data integrity and privacy

The security of the data accessed by an application is secured at origin, destination, and in transit:

- **Integrity** Data sent over the network can not be altered in transit.
- **Privacy/confidentiality** Data sent over the network cannot be intercepted and read in transit.

Identity propagation

The final element of security is one of propagation of identity:

- **Single sign on** When a client request needs to flow through several systems within the enterprise, the client is not forced to provide authentication data multiple times. The single sign on method is used to propagate the authentication information to downstream systems that can in turn apply access control.

Authentication

When global security is turned on, clients must be authenticated.

If a client tries to access a secured application without being authenticated, an exception is generated.

Table 3 lists typical clients that would invoke WebSphere Process Server components, and the authentication options available for each type of client.

Table 3. Authentication options for various clients

Client	Authentication options	Notes
Web services clients	You can use WS-Security/SOAP authentication.	
Web or HTTP clients	HTTP Basic authentication (the browser prompts the client for a user name and password).	These clients reference JSPs, Servlets, and HTML documents.
Java clients	JAAS.	
All clients	SSL client authentication.	

Some of the components of the WebSphere Process Server infrastructure have authentication aliases that are used to authenticate the runtime code for access to databases and the messaging engine. These Business Process Choreographer and

Common Event Infrastructure authentication aliases are outlined in subsequent topics. The WebSphere Process Server installer collects the user name and passwords to create these aliases.

Some runtime components have message-driven beans (MDBs) that are configured with a runAs role. The WebSphere Process Server installer collects the user name and password for the runAs role.

Modifying authentication aliases:

You might need to modify existing authentication aliases.

Modify authentication aliases from the administrative console.

1. Access the J2EE Connector Architecture (J2C) authentication data entries panel.

   From the administrative console, expand **Security**, and click **Global security**. Under the Authentication heading, expand **JAAS Configuration**, and select **J2C Authentication data**.

   Note: You can also use this panel to add or delete authentication data.
2. Select the authentication alias that you want to modify.

   The J2EE Connector Architecture (J2C) authentication data entries panel contains a list of authentication aliases, the user ID associated with this alias, and optionally a description of the alias. Click on the alias that you want to modify.
3. Change the properties of the alias.

   On the configuration panel for the selected alias, you can modify either the alias name or the associated user ID and password. You can also modify the description of the authentication data entry.
4. Confirm your changes.

   Click either **OK** or **Apply**. Return to the J2EE Connector Architecture (J2C) authentication data entries panel, and click **Save** to apply your changes to the master configuration.

   Note: For a Network Deployment installation, make sure that a file synchronize operation is performed to propagate the changes to other nodes.

   For related information see *Augmenting WebSphere Process Server profiles with security*

Access control

Access control refers to ensuring that an authenticated user has the permissions necessary to access resources or to perform a specific operation.

When a general user is authenticated to theWebSphere Process Server it is important for security that not every possible operation is available to that user. Allowing some users to perform certain tasks, while denying these tasks to other users is termed access control.

Access control can be arranged for components that you develop to make them secure. You do this by using service component architecture qualifiers at development time. See the WebSphere Integration Developer Information Center for more information.

Some WebSphere Process Server components, packaged as enterprise archive (EAR) files, secure their operation using J2EE role-based security. Details of these components are provided. The Business Process Choreographer and the Common Event Infrastructure are installed as part of WebSphere Process Server. The role-based security associated with these components is outlined in detail in subsequent topics.

Data integrity and privacy

The privacy and integrity of data that is accessed when WebSphere Process Server processes are invoked is critical to your security.

Data privacy and data integrity are closely related concepts.

Privacy

Privacy means that it should not be possible for an unauthorized user to intercept and read data.

Integrity

Integrity means that it should not be possible for an unauthorized user to alter data.

Solutions provided in WebSphere Process Server

WebSphere Process Server supports two widely used solutions for data privacy and integrity:

- Secure Sockets Layer (SSL) protocol. SSL uses a handshake to authenticate the end points and exchange information that is used to generate the session key that will be used by the end points for encryption and decryption. SSL is a synchronous protocol and is suitable for point to point communication. SSL requires that the two end points maintain a connection with each other for the duration of the SSL session.
- WS-Security. This standard defines Simple Object Access Control (SOAP) extensions for securing SOAP messages. WS-Security adds support for authentication, integrity, and privacy for a single SOAP message. Unlike SSL, there is no handshake to establish a session key. This makes WS-Security suitable for securing messages in an asynchronous environment, such as SOAP over Java Message Service (JMS) or SOAP over Service Integration Bus (SIB).

In a business integration environment with multiple systems interacting with one another, it is likely that some of the communication will be asynchronous. Therefore, in most instances, WS-Security is the superior solution.

Single sign on

A client is asked to provide user name and password information only once. The provided identity propagates throughout the system.

When a client request needs to flow through multiple systems within the enterprise, the client needs to authenticate only once. This concept of identity propagation is solved using a single sign on method.

The authenticated context is propagated to downstream systems, which can apply access control.

Either Tivoli Access Manager WebSEAL or Tivoli Access Manager plug-in for Web servers can be used as reverse proxy servers to provide access management and single sign on capability to WebSphere Process Server resources. Details of how to configure these tools can be found in the WebSphere Application Server documentation.

Related information

Configuring single sign-on capability with Tivoli Access Manager or WebSEAL

Default security of installed components

Several important components of WebSphere Process Server have default security information. This information includes aliases to which default users are mapped and security roles to which users must be granted access in order to invoke these components.

Purpose

Several of the important components of WebSphere Process Server use pre-defined aliases for authenticating with messaging engines and databases. The user names and passwords in the applicable response file are associated with these aliases.

Business Process Choreographer authentication aliases

Business processes have the following authentication aliases. Modify these aliases using the administrative console.

The aliases in Table 4 are used to invoke the components regardless of the identity of the invoking user.

Table 4. Authentication aliases associated with business processes.

Alias	Description	Information
BPEAuthDataAliasJMS_ *node_server* A character space has been added to this entry to enable it to fit in the table cell. The actual alias name does not contain a character space.	Used to authenticate with the messaging engine.	Enter user name and password values in the applicable Business Process Choreographer properties in the response file.
BPEAuthDataAlias *DbType_node_server* A character space has been added to this entry to enable it to fit in the table cell. The actual alias name does not contain a character space.	Used to authenticate with databases.	Configure the database using the provided scripts.

Table 5 describes the RunAs roles created for business processes.

Table 5. RunAs roles associated with business processes.

RunAs role	Description	Information
JMSAPIUser	Used by the BFM JMS API MDB in bpecontainer.ear.	Enter user name and password values in the applicable Business Process Choreographer properties in the response file.

Table 5. RunAs roles associated with business processes. (continued)

RunAs role	Description	Information
EscalationUser	Used by the task.ear MDB.	Enter user name and password values in the applicable Business Process Choreographer properties in the response file.

The user name that you supply will be added to the RunAs role.

Common Event Infrastructure authentication aliases

The Common Event Infrastructure has the following authentication aliases. Modify these aliases using the administrative console.

The aliases in Table 6 are used to invoke the components regardless of the identity of the invoking user.

Table 6. Authentication aliases associated with the Common Event Infrastructure.

Alias	Description	Information
CommonEventInfrastructure JMSAuthAlias A character space has been added to this entry to enable it to fit in the table cell. The actual alias name does not contain a character space.	Used to authenticate with the messaging engine.	Enter user name and password values in the applicable Common Event Infrastructure configuration properties in the response file.
EventAuthAlias*DBType*	Used to authenticate with databases.	Enter user name and password values in the applicable Common Event Infrastructure configuration properties in the response file.

Service component architecture authentication aliases

The service component architecture (SCA) has the following authentication aliases. Modify these aliases using the administrative console.

The aliases in Table 7 are used to invoke the components regardless of the identity of the invoking user.

Table 7. Authentication aliases associated with SCA components.

Alias	Description	Information
SCA_Auth_Alias	Used to authenticate with the messaging engine.	Enter user name and password values in the applicable SCA configuration properties in the response file.

Access control in business process and human task applications

The following enterprise archive (EAR) files are installed with access control as part of the Business Process Choreographer installation. Business Process

Choreographer is installed as part of the WebSphere Process Server installation. The human task manager uses the roles to determine the capabilities of the user on a production system.

EAR file	Roles	Default permission	Notes
bpecontainer.ear	BPESystemAdministrator	Group name entered during the installation.	Has access to all business processes and all operations.
bpecontainer.ear	BPESystemMonitor	All authenticated users.	Has access to read operations.
task.ear	TaskSystemAdministrator	Group name entered during the installation.	Has access to all human tasks.
task.ear	TaskSystemMonitor	All authenticated users.	Has access to read operations.
Bpcexplorer.ear	WebClientUser	All authenticated users.	Can access the Business Process Choreographer Explorer.

Access control in Common Event Infrastructure applications

The following enterprise archive (EAR) file is installed with access control as part of the Common Event Infrastructure installation. The Common Event Infrastructure is installed as part of the WebSphere Process Server installation.

The EventServer.ear file is the only EAR file installed as part of the Common Event Infrastructure installation.

Roles	Default permission
eventAdministrator	All authenticated users.
eventConsumer	All authenticated users.
eventUpdater	All authenticated users.
eventCreator	All authenticated users.
catalogAdministrator	All authenticated users.
catalogReader	All authenticated users.

Securing applications in WebSphere Process Server task roadmap

The applications that you deploy to your WebSphere Process Server instance require security to be built in to them and to be applied at runtime.

Securing your applications assumes that you have global security enabled.

The applications that you host in your WebSphere Process Server environment perform many business critical functions that require security. Some applications will access, transfer or alter sensitive information (for example: payroll information or credit card details). Others will perform billing or inventory management. Naturally the security of these applications is vitally important.

Secure your applications by performing the following tasks:

1. Develop your applications in WebSphere Integration Developer using all appropriate security features.
2. Deploy your applications to your WebSphere Process Server environment assigning users or groups to appropriate security roles.
3. Maintain the security of your WebSphere Process Server environment.

Developing secure components

Secure the components that you develop. Components implement interfaces that have methods. Use the service component architecture (SCA) qualifier SecurityPermission to secure an interface or method.

Develop your secured application in WebSphere Integration Developer. Export the application as an enterprise archive (EAR) file for deployment in WebSphere Process Server.

Import a secured application into WebSphere Process Server with the following steps.

1. Install the application EAR file.

   On the administrative console, expand **Applications** and click **Enterprise applications**. Click **Install** and fill in the details of the new application.
2. Assign security roles to the new application.

   Click **Map security roles to users/groups**. You have four choices of roles for the application.

Option	Description
Everyone	This is equivalent to no security.
All authenticated	Anyone who authenticates with a valid user name and password is a member of the role.
Mapped users	Individual users are listed as members of the role.
Mapped groups	Groups are the most convenient way to add the users, every member of the identified groups becomes a member of the role.

   Use **Look up users** and **Look up groups** to list users and groups that can be mapped to the role.

In the sample SCDL below, access to the method **onewayinvoke** is restricted to users that are members of the **manager** role.

```
<?xml version="1.0" encoding="UTF-8"?>
<scdl:component xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:java="http://www.ibm.com/xmlns/prod/websphere/scdl/java/6.0.0"
xmlns:ns1="http://sample.recovery.security/Itarget"
xmlns:scdl="http://www.ibm.com/xmlns/prod/websphere/scdl/6.0.0"
xmlns:wsdl="http://www.ibm.com/xmlns/prod/websphere/scdl/wsdl/6.0.0"
displayName="secure" name="Component1">
  <interfaces>
    <interface xsi:type="wsdl:WSDLPortType" portType="ns1:Itarget">
      <method name="onewayinvoke">
       <scdl:interfaceQualifier xsi:type="scdl:SecurityPermission"
role="manager"/>
      </method>
      </interface>
  </interfaces>
  <references/>
```

```
  <implementation xsi:type="java:JavaImplementation"
class="sca.component.java.impl.Component1Impl1">
 </implementation>
</scdl:component>
```

Deploying (installing) secure applications

Deploying applications that have security constraints (secured applications) is similar to deploying applications with no security constraints. The only difference is that you might need to assign users and groups to roles for a secured application, which requires that you have the correct active user registry. If you are installing a secured application, roles would have been defined in the application. If delegation was required in the application, RunAs roles also are defined and a valid user name and password must be provided.

Before you perform this task, verify that you have designed, developed, and assembled an application with all the relevant security configurations. For more information about these tasks see the WebSphere Integration Developer information center. In this context, deploying and installing an application are considered the same task.

One of the required steps to deploy secured applications is to assign users and groups to the roles that were defined when the application was constructed. This task is completed as part of the step entitled, ″Map security roles to users and groups″. If an assembly tool was employed, this assignment may have been completed in advance. In that case you can confirm the mapping by completing this step. You can add new users and groups and modify existing information during this step.

If a RunAs role has been defined in the application, the application will invoke methods using an identity setup during deployment. Use the RunAs role to specify the identity under which the downstream invocations are made. For example, if the RunAs role is assigned user, ″bob″, and the client, ″alice″, is invoking a servlet, with delegation set, which calls the enterprise beans, the method on the enterprise beans is invoked with ″bob″ as the identity. As part of the deployment process one of the steps is to assign or modify users to the RunAs roles. This step is entitled, ″Map RunAs roles to users″. Use this step to assign new users or modify existing users to RunAs roles when the delegation policy is set to SpecifiedIdentity.

The steps described below are common for both installing an application and modifying an existing application. If the application contains roles, you see the ″Map security roles to users and groups″ link during application installation and also during managing applications, as a link in the Additional properties section.

1. In the administrative console expand Applications and click Install New Application.

   Complete the steps that are required for installing applications prior to the step entitled, ″Map security roles to users and groups″.
2. Assign users and groups to roles.
3. Map users to RunAs roles if RunAs roles exist in the application.
4. Click Correct use of System Identity to specify RunAs roles, if needed.

   Complete this action if the application has delegation set to use system identity, which is applicable to enterprise beans only. System identity uses the WebSphere Process Server security server ID to invoke downstream methods. use this ID with caution because this ID has more privileges than other identities in accessing WebSphere Process Server internal methods. This task is provided to make sure that the deployer is aware that the methods listed in the

panel have system identity set up for delegation and to correct them if necessary. If no changes are necessary, skip this task.

5. Complete the remaining non-security related steps to finish installing and deploying the application.

After a secured application is deployed, verify that you can access the resources in the application with the correct credentials. For example, if your application has a protected Web module, make sure only the users that you assigned to the roles can use the application.

Related information

Assigning users and groups to roles

Assigning users to RunAs roles

Assigning users to roles

A secured application uses one or both of the security qualifiers securityPermission and securityIdenitity. When these qualifiers are present there are additional steps which must be taken at deployment time in order that the application and its security features work correctly.

This task assumes that you have a secured application ready to deploy as an EAR file into WebSphere Process Server.

Applications implement interfaces that have methods. You can secure an interface or a method with the service component architecture (SCA) qualifier securityPermission. When you invoke this qualifier you specify a role (for example, “supervisors”) that has permission to invoke the secured method. When you deploy the application you have the opportunity to assign users to the specified role.

The securityIdentity qualifier is equivalent to the RunAs role used for delegations in WebSphere Application Server. The value associated with this qualifier is a role. During deployment the role is mapped to an identity. Invocation of a component secured with securityIdentity takes the specified identity, regardless of the identity of the user that is invoking the application.

1. Follow the instructions for deploying an application into WebSphere Process Server. See *Installing a module on a production server* for more details.
2. Associate the correct users with the roles.

Security qualifier	Action to take
securityPermission	Assign a user or users to the role specified. There are four choices: • Everyone - equivalent to no security. • All authenticated - every authenticated user is a member of the role. • Mapped User - Individual users are added to the role. • Mapped Groups - Groups of users are added to the role. The most flexible choice is Mapped Groups, because users can be added to the group and thus gain access to the application without restarting the server.

Security qualifier	Action to take
securityIdentity	Provide a valid user name and password for the identity to which the role is mapped.

Related information

Delegations

Security and the Common Event Infrastructure

You can use WebSphere method-level declarative security to restrict access to Common Event Infrastructure functions.

The Common Event Infrastructure defines six security roles, each one associated with a related group of functions. These security roles control access to both programming interfaces and commands. (The default configuration of the Common Event Infrastructure does not require the use of these roles; however, in a Network Deployment environment, the WebSphere Process Server needs to be authenticated with the same users assigned to the Common Event Infrastructure security roles. For more information about security roles, see *Learning about security* and *Role-based authorization* in the WebSphere Application Server Information Center.) If you are already a WebSphere Process Server authenticated user, and global security is turned on, you can access the Common Event Infrastructure resources.

Note:

If the security roles are used by mapping specific users to the roles, the authenticated users need to be the same users as assigned to the security role. For additional information about authenticated users and the `RunAs` role, see *Assigning users to RunAs roles*.

The following table describes the security roles and the types of users associated with each role.

Table 8. Security roles and user types

Security role	User types
eventAdministrator	Event consumers that need to query, update, and delete events stored in the event database. This role provides access to the following interfaces: • EventAccess.purgeEvents() • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • EventAccess.updateEvents() • Emitter.sendEvent() • Emitter.sendEvents() • eventquery.jacl • eventpurge.jacl • emitevent.jacl • eventbucket.jacl

Table 8. Security roles and user types (continued)

Security role	User types
eventConsumer	Event consumers that need to query events stored in the event database. This role provides access to the following interfaces: • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • eventquery.jacl
eventUpdater	Event consumers that need to update events stored in the event database. This role provides access to the following interfaces: • EventAccess.updateEvents() • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • eventquery.jacl
eventCreator	Event sources that need to submit events to an emitter using synchronous EJB calls. This role provides access to the following interfaces: • Emitter.sendEvent() • Emitter.sendEvents() • emitevent.jacl **Note:** The eventCreator role restricts access to event submission only if the emitter is configured to use synchronous EJB calls for event transmission. If the emitter uses asynchronous JMS messaging for event transmission, you must use JMS security to restrict access to the destination used to submit events.
catalogAdministrator	Event catalog applications that need to create, update, delete, or retrieve event definitions in the event catalog. This role provides access to all methods of the EventCatalog interface and all functions of the **eventcatalog.jacl** script. Because changes to the event catalog can result in generation of events, this role also provides access to event submission interfaces.
catalogReader	Event catalog applications that need to retrieve event definitions from the event catalog. This role provides access to the following interfaces: • EventCatalog.getAncestors() • EventCatalog.getChildren() • EventCatalog.getDescendants() • EventCatalog.getEventDefinition() • EventCatalog.getEventDefinitions() • EventCatalog.getEventExtensionNamesForSourceCategory() • EventCatalog.getEventExtensionToSourceCategoryBindings() • EventCatalog.getParent() • EventCatalog.getRoot() • EventCatalog.getSourceCategoriesForEventExtension() • eventcatalog.jacl (-listdefinitions option) • eventcatalog.jacl (-listcategories option) • eventcatalog.jacl (-exportdefinitions option)

Note:

The security roles most relevant to utilizing the functionality of the Common Event Infrastructure are **eventAdministrator** and **eventConsumer** .

The event server message-driven bean runs using the WebSphere Process Server user identity. If you are using asynchronous JMS transmission to submit events to the event server, and you have enabled method-based security, you must map this user identity to the eventCreator role.

Note:

If Java 2 security is enabled, you must modify your policy file to enable access to certain functions:

- If you are running an event source application and you want to generate your own globally unique identifiers (GUIDs), add the following entries:

```
permission java.io.FilePermission "${java.io.tmpdir}${/}guid.lock",
  "read, write, delete";
permission java.net.SocketPermission "*", "resolve";
```

- If you are using the default filter plug-in or the notification helper to filter events using XPath event selectors, add the following entries:

```
permission java.util.PropertyPermission "*", "read";
permission java.io.FilePermission
  "${was.install.root}${/}java${/}jre${/}lib${/}jxpath.properties",
  "read";
```

Securing adapters

Two types of adapter are supported in WebSphere Process Server: WebSphere Business Integration Adapters and WebSphere Adapters. The security of both types of adapter is discussed.

Adapters are the mechanism by which applications communicate with Enterprise Information Systems (EISs). The information that is exchanged between an application and an EIS can be highly sensitive. It is important to ensure the security of this information transaction.

WebSphere Business Integration Adapters consist of a collection of software, application program interfaces (APIs) and tools that enable applications to exchange business data through an integration broker. WebSphere Business Integration Adapters rely on JMS messaging and JMS does not support security context propagation.

WebSphere Adapters enable managed, bidirectional connectivity between Enterprise Information Systems (EISs) and J2EE components supported by WebSphere Process Server.

For inbound communication from both types of adapter into WebSphere Process Server, there is no authentication mechanism. For WebSphere Business Integration Adapters the reliance on JMS messaging precludes security context propagation. J2C also lacks inbound security support, therefore WebSphere Adapters also have no authentication mechanism for inbound communication.

The entry from an adapter to WebSphere Process Server always employs a service component architecture (SCA) export. The SCA export has to be wired to an SCA component, such as mediation, business process, SCA Java component or Selector.

The security solution is to define a runAs role on the component that is the target for the WebSphere Adapter export. This is done using the SCA qualifier SecurityIdentity during development (see the WebSphere Integration Developer Information Center for more information). When the component runs, it does so under the identity defined in the runAs role.

The value for SecurityIdentity is a role not a user. Nevertheless, when the EAR file is deployed to WebSphere Process Server you must provide a user name and password for the identity that is to be used. The use of SecurityIdentity prevents exceptions being thrown if a downstream component is secured and requires the client to have an authenticated identity.

Note: The use of SecurityIdentity does not secure the communication between the adapter and the EIS.

WebSphere Business Integration Adapters send data to WebSphere Process Server as JMS messages over the service integration bus.

WebSphere Adapters reside in the JVM of the WebSphere Process Server, and therefore only the communication between the adapter and the target EIS needs to be secured. The protocol between the adapter and the EIS is EIS-specific. The documentation of the EIS will provide information about how to secure this link.

Related concepts

Security considerations for service integration buses

Security in human tasks and business processes

There are a number of roles associated with human tasks and business processes. This topic describes the roles available.

Human tasks, by definition, require human intervention to complete them. Some business processes might also require human intervention. These human tasks and business processes are developed using WebSphere Integration Developer and are invoked using Business Process Choreographer. When you develop the task or process, you must assign roles to users or groups involved in the human tasks and business processes. See the WebSphere Integration Developer Information Center for more information about assigning the roles or querying the roles associated with specific roles.

The Human Task Manager uses the roles to determine the users' capabilities on a production system.

Roles associated with tasks and processes

Important: These roles are unique to tasks and processes that are running in the Business Process Choreographer business container and human task container.

WebSphere Process Server supports the following roles for tasks and processes:

Administrator
Users who belong to this role can monitor, end, or delete tasks and processes and also display information about tasks and processes.

Reader
Users who belong to this role can only display tasks and processes.

Starter
Users who belong to this role can start or display tasks and processes.

Tasks also have these additional roles:

Owner
Users who belong to this role can save, cancel, complete or display tasks that they have already claimed.

Potential owner
Users who belong to this role can claim and display tasks.

Securing the WebSphere ESB

If you want to ensure the confidentiality and integrity of messages in transit across an enterprise service bus, you need to enable bus security, so that access to the bus itself and to all resources on the bus must be authorized.

To enable bus security, you enable WebSphere global security and enable messaging security for each service integration bus that forms part of the enterprise service bus. For more information about enabling bus security, see Messaging security.

Bus security comprises a number of components that can be used together to ensure that users are authenticated, that resources are protected by security checks, and that messages are secure when they are in transit from a service requester to a service provider. Security covers all of the following areas:

- Authentication and authorization of users, when connecting to a service integration bus and when using the bus resources.
- Secure communication transports between the client and messaging engine and between messaging engines.
- Authentication of messaging engines joining a bus.
- Authentication of message store (database) users.

WebSphere ESB provides the authentication alias SCA_Auth_Alias that you can use for authentication.

These topics are supplemental to the core security information provided for WebSphere Application Server. For a more detailed discussion of securing your environment and applications, see the Securing applications and their environments.

Creating end to end security

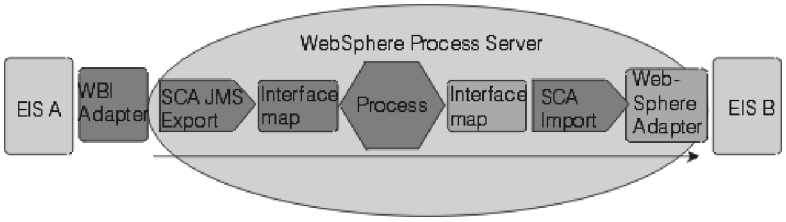
There are many potential end to end security scenarios. Each of these might involve differing security steps. Several typical scenarios, with the necessary security options, are presented.

These scenarios all assume that global security is enforced.

1. Determine which of the examples provided in this section, most closely match your security needs. In certain instances, your scenario will involve a combination of information from more than one of the examples.
2. Read the security information for the relevant scenarios and apply it to your security needs.

Classic integration scenario - inbound and outbound adapters

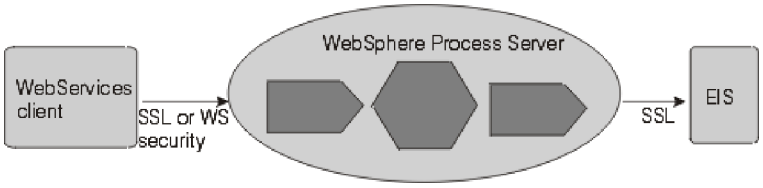
An inbound request comes in from a WebSphere Business Integration Adapter. The service component architecture (SCA) invokes an interface map based on the SCA export. The request flows through a process component, a second interface map and is then passed on to a second EIS (B), via a WebSphere Adapter. These are SCA invocations, with one component invoking a method on the next component.



There is no authentication mechanism for the inbound adapter. You can establish the security context by defining the SecurityIdentity qualifier on the first component - in this instance, the first interface map component. From that point, SCA will propagate the security context from each component to the next. Access control for each component is defined by use of the SecurityPermission qualifier.

Inbound Web service request to WebSphere Process Server

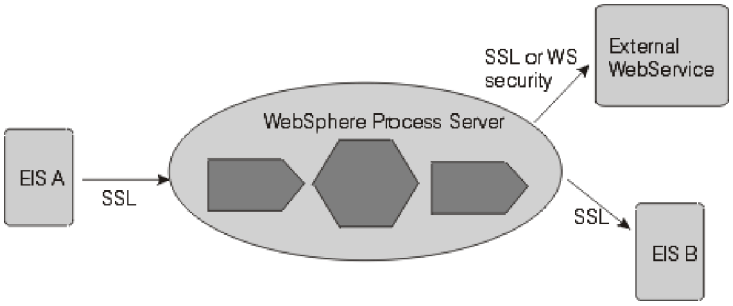
In this scenario a Web service client invokes a WebSphere Process Server component. The request passes through several components in the WebSphere Process Server environment before being passed to an EIS by an adapter.



You can authenticate the Web service client as a SSL client, using HTTP Basic authentication or using WS-Security authentication. When the client is authenticated, access control is applied based on the SecurityPermission qualifier. Between the client and the WebSphere Process Server instance, you can secure the data integrity and privacy using SSL or WS-Security. SSL secures the entire pipe, whereas with WS-Security you can encrypt or digitally sign parts of the SOAP message. For Web services, WS-Security is the preferred standard.

Outbound Web service request from WebSphere Process Server

In this scenario the inbound request can be from an adapter, a Web service client, or a HTTP client. WebSphere Process Server a component (for instance a BPEL component) invokes an external Web service.



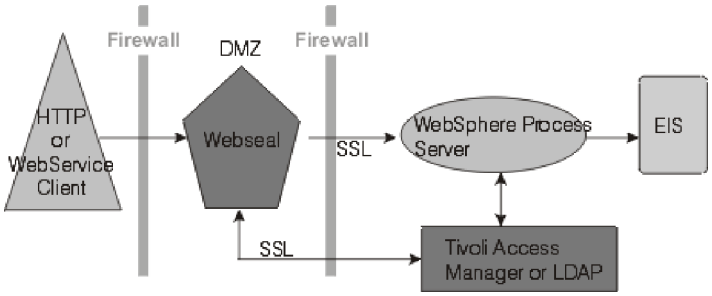
As for the inbound Web service request, you can authenticate with the external Web service as a SSL client, using HTTP Basic authentication or using WS-Security authentication. Use LTPACallBackHandler as the callback mechanism to extract the usernameToken from the current RunAs subject. Between WebSphere Process Server and the target Web service, you can ensure data privacy and integrity using WS-Security.

Web application - HTTP inbound request to WebSphere Process Server

WebSphere Process Server supports three types of authentication for HTTP:

- HTTP basic authentication
- HTTP forms based authentication
- HTTPS SSL-based client authentication.

In addition, to protect your intranet from intruders, you can place the Web server in the demilitarized zone (DMZ), and the WebSphere Process Server inside the inner firewall. In this example, Webseal is used as the reverse proxy, which performs the authentication. It has a trust association with WebSphere Process Server behind the firewall and can forward authenticated requests.



Related concepts

Security considerations for service integration buses

Tutorial: Writing a Jacl script that lists security roles

This tutorial addresses how to write and execute a simple Jacl script that can access and manage a JMX MBean. This particular script is concerned with calling roles when global security is enables. Using this script, you will be able to print out the role name for each role in a relationship.

Objective of this tutorial

After completing this tutorial, you will be able to:

- Write a Jacl script that calls a JMX MBean requesting a list of all relationships.

For more information about writing scripts, refer to ″Using scripting (wsadmin)″ in the WebSphere Application Server Network Deployment, version 6.0 information center.

Time required to complete this tutorial

This tutorial requires approximately 15-30 minutes to complete.

Prerequisites

This tutorial uses a script that is included with the JMX Security sample. This sample demonstrates the MBean function of printing out a list of role relationships.

Note: To use this script, you must select the option to install code samples during the installation of WebSphere Process Server.

The location of the sample Jacl script is in <wbi_root>/samples/JMXSample/scripts. The name of the script is: `RelServicesAdmin.jacl`.

To run the script, enter:

```
wsadmin -f ../samples/JMXSample/scripts/RelServicesAdmin.jacl
 -server servername -node nodename
```

This script will call up to 10 relationships in your environment and up to 10 roles for each relationship will be printed on the console.

Exercise: Writing a Jacl script

The basic concepts in this script can be used to communicate with any MBean in the system. All that is required is the name and type of the MBean and the methods and attributes available on the MBean. The getAttribute and setAttribute commands are used for attributes. The invoke command is used for methods. Follow these steps to create a `.Jacl` script that manages the JMX Security MBean.

Note: The code in each step is prefaced with a statement explaining what the code does.

1. Determine the **nodename**

   The first part of the script shown below determines the nodename. If the `nodeName` is not specified correctly, the correct syntax is printed and the script exits.

   ```
   # read and validate arguments

       if {{$argc == 1 } && { [lindex $argv $i] == "-nodeName" } {
               set nodeName [lindex $argv $i]
   ```

2. Identify the **MBean**

   An MBean is identified by a type and a name.

   Note: The name and type are hard coded in this case since you know the specific MBean you want to use.

   The second part of the script identifies the MBean.

   ```
   # these two variables, mbeanName and mbeanType are used
     to uniquely identify the mbean.
          # for this sample, the mbean that access relationship
   ```

```
        services will be used.

      set mbeanName"RelService"
      set mbeanType"WBIRelServices"
```

3. Locate and set the **reference** to the MBean.

   You use the code shown here to set the reference for the MBean.

```
# locate the mbean and set a reference to it in "relSvcsMBean" variable

      set relSvcsMBean [$AdminControl queryNames
          name=$mbeanName,node=$nodeName,type=$mbeanType,*]
```

4. Call the **relationship** using the getAttribute command.

   The documentation of this specific MBean defines an attribute named `allRelationshipNames`. Ask the MBean for that attribute using the getAttribute command. The attribute value will be a list that you step through in the next step that invokes the command.

```
# request the list of relationships from the mbean

      set relationships
       [$AdminControl getAttribute $relSvcsMBean allRelationshipNames]
```

5. Invoke the **command** for each relationship name, you print the name, and then go back to the MBean for additional information.

   In this example the MBean defines a method named getAllRoleNames with a single parameter for the specific relationship name. You use the invoke command to call this method, passing the current relationship name. For each role in the relationship, a role name is printed.

```
# loop through the list of role names and print name

        foreach roleName $roles {
          puts "    Role: $roleName"
        }
    }
} else {
      # arguments were not correct, print correct syntax
         puts "Usage: wsadmin -f RelServicesAdmin.jacl -nodeName nodeName"
}
```

You have now written a script to call relationships.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both: IBM, IBM (logo), AIX, CICS, Cloudscape, DB2, DB2 Connect, DB2 Universal Database, developerWorks, Domino, IMS, Informix, iSeries, Lotus, MQSeries, MVS, OS/390, Passport Advantage, pSeries, Rational, Redbooks, Tivoli, WebSphere, z/OS, zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (http://www.eclipse.org/).

IBM Websphere Process Server for z/OS version 6.0.1

Printed in USA