WebSphere® WebSphere Process Server for z/OS

IBM

Version 6.1.0

Securing Applications and Their Environments

WebSphere. WebSphere Process Server for z/OS

Version 6.1.0

Securing Applications and Their Environments

Note

Before using this information, be sure to read the general information in the Notices section at the end of this document.

31 March 2008

This edition applies to version 6, release 1, modification 0 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, send an e-mail message to 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.

1. The information presented here is available in Adobe® PDF format at the following link: WebSphere Process Server documentation (in PDF format).
2. Business Process Management information roadmaps on IBM® developerWorks® organize information about WebSphere Process Server and the other products in the portfolio.

These documents are supplemental to the core security documentation located in the WebSphere Application Server Information Center and specifically in the WebSphere Application Server Security Documentation.

The security of your data and processes is critical. WebSphere Process Server security is based on the WebSphere Application Server version 6.1 security. Refer to the WebSphere Application Server Information Center for detailed information about security.

General overview

The security of your data and processes is critical.

WebSphere Process Server security is based on the WebSphere Application Server version 6.1 security. Refer to the WebSphere Application Server Network Deployment, version 6 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 administrative security, enabling application security, creating profiles with security, and restricting access to critical functions to selected users.

There are several aspects to securing an application. These include:

- **“Authentication” on page 32**; a user or a process that invokes an application must be authenticated.
- **“Access control” on page 33**; does the authenticated user have permission to perform the operation?
- **“Data integrity and privacy” on page 34**; the data that is accessed by an application must be secured so that no unauthorized party can view or modify it in any way.
- **“Single sign on” on page 35**; 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.

Security considerations specific to WebSphere Process Server

WebSphere Process Server security is built on WebSphere Application Server 6.1 security. Considerations which are specific to WebSphere Process Server are listed.

WebSphere Process Server security features

- The administrative console panel Business Integration Security is unique to WebSphere Process Server. It can be reached by expanding **Security** and clicking **Business Integration Security**. This panel allows users to assign specific identities from their user registry to important business integration authentication aliases. In addition you can administer you Business Process Choreographer security settings on this panel.
- Application security is turned on by default in WebSphere Process Server. This is not the case in WebSphere Application Server.
- There are a set of component-specific security roles.

Getting started with security

Security is an integral consideration when planning to install WebSphere Process Server, when developing and deploying applications and in the day-to-day running of your process server.

About this task

To maintain the security of your sensitive data, you must secure both the process server environment and the applications that you deploy to that environment.

Procedure

1. Consider security when you install WebSphere Process Server. See “Installing WebSphere Process Server: security considerations” on page 3
2. Ensure that security is turned on for your standalone, or deployment environment installation.
   a. Ensure that “Administrative security” on page 8 is turned on. Administrative security is turned on by default.
   b. Ensure that “Application security” on page 9 is turned on. Application security is turned on by default.
   c. If required turn on “Java 2 security” on page 10.
   d. Use the Security Configuration wizard in the administrative console to configure security options.
   e. Set up a secure authentication mechanism and user account repository.
   f. Assign user names and passwords to important business integration authentication aliases.
   g. Assign users to appropriate administrative security roles.
3. Secure the applications that you deploy to your process server environment.
   a. Develop your applications in WebSphere Integration Developer using all appropriate security features.
   b. Deploy your applications to your WebSphere Process Server environment.
   c. Assign users or groups to appropriate security roles to control access to the newly deployed application.
   d. Maintain the security of your WebSphere Process Server environment.

Installing WebSphere Process Server: security considerations

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

About this task

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

Procedure

1. Secure your environment before installation.

   The commands required to install WebSphere Process Server with proper security will depend on your operating system. For detailed information about steps to take before installing, see the topic **Preparing for security at installation time** in the WebSphere Application Server for z/OS Information Center.
2. Prepare the operating system for installation of WebSphere Process Server.

   For detailed information about preparing your z/OS® operating system for installation, see the topic **Preparing the base operating system**Preparing the base operating system 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 and configure WebSphere Process Server for z/OS. For detailed information about securing your environment after installing and configuring, see the topic **Securing your environment after installation**Securing your environment after installation in the WebSphere Application Server Information Center.

What to do next

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

Related information

Securing your environment before installation

Preparing the operating system for product installation

Securing your environment after installation

Authentication information provided at install time

In previous releases of WebSphere Process Server you were prompted for various authentication information during installation. Now all the components default to the primary administrative credentials that you provide. These default values provide basic security, but in order to harden the security of your installation you should use the administrative console to configure the various components of WebSphere Process Server to have appropriate security identities.

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 administrative security. You can use either the user registry of the local operating system (this is the default) or the Lightweight Directory Access Protocol (LDAP).

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.

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.

About this task

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.

You must hold the edited response files in a secure location because the user names and passwords are stored in plain text.

Procedure

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 (**scaSecurityUserid** and **scaSecurityPassword**).
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 (**ceiSampleJmsUser** and **ceiSampleJmsPwd**).
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.

   Enter a valid user name and password in the **bpcmqUser** and **bpcmqPwd** fields.

What to do next

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

Related tasks

"Modifying authentication aliases" on page 33
You might need to modify existing authentication aliases.

Configuring WebSphere Process Server security for a standalone server

Follow the links below for how to configure the security of a standalone installation of WebSphere Process Server.

Securing a stand-alone WebSphere Process Server installation

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.

Before you begin

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

About this task

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.

Procedure

1. Ensure that administrative security is turned on. "Enabling administrative security" on page 7.
2. Ensure that application security is turned on. "Securing applications in WebSphere Process Server" on page 31.
3. Add users or groups to the administrative role. You can give administrative rights to individual users or to a group of users by following the **Administrative User Roles** or **Administrative Group Roles**, respectively.
4. Select the user account repository that 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
Federated repositories	Specify this setting to manage profiles in multiple repositories under a single realm. The realm can consist of identities in: • The file-based repository that is built into the system, • One or more external repositories, • Both the built-in, file-based repository and in one or more external repositories. **Note:** Only a user with administrator privileges can view the federated repositories configuration. For more information, see Managing the realm in a federated repository configuration.
Local Operating System	The default user registry. For details about how to configure the user account registry, see "Configuring the user account repository" on page 11.
Standalone LDAP registry	To configure LDAP as your user registry, follow the instructions in Configuring Lightweight Directory Access Protocol (LDAP) as the user registry.
Standalone custom registry	For details of how to configure the user account registry, see "Configuring the user account repository" on page 11.

5. Apply these changes.

   Click the **Apply** button at the bottom of the panel.
6. Go to the Business Integration Security panel. Expand **Security** and click **Business Integration Security**.
7. Supply appropriate user identities for the listed authentication aliases. The credential you provide must exist in the user account repository that you are employing.
8. On the same panel you can configure security for Business Process Choreographer.

   Set the business process choreographer user role mappings for the business flow and human task manager:
   - **Administrator**: User name(s) and/or group name(s) for the business flow and human task administrator role. Users assigned to this role have all privileges.
   - **Monitor**: User name(s) and/or group name(s) for the business flow and human task monitor role. Users assigned to this role can view the properties of all of the business process and task objects.

   The business process choreographer authentication aliases can be configured for each deployment target where the business process choreographer has been installed. The following authentication aliases are listed:
   - **JMS API Authentication**: Authentication for the business flow manager message-driven bean to process asynchronous API calls.
   - **Escalation User Authentication**: Authentication for the human task manager message-driven bean to process asynchronous API calls.
9. Apply these changes.

   Click the **Apply** button at the bottom of the panel.

10. Save the changes to the local configuration.

    Click **Save** in the message pane.

11. 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.

Results

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

Each node that you create must be secured in this way. The system administrator user identity may have been used in multiple places during installation and configuration of environment. It is advisable to replace this identity with appropriate user credentials from the user account repository for all but the core security functions. Use the **Business Integration Security** panel in the administrative console to administer these identities and aliases.

Enabling administrative security

In WebSphere Application Server version 6.1, administrative security is enabled by default. If you have disabled administrative security, the following instructions describe how to enable it again.

Before you begin

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

About this task

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.

Procedure

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

   Expand **Security** and click **Secure administration, applications, and infrastructure**.

2. Enable administrative security.

   Select **Enable administrative security**.

3. Optional: Enforce Java™ 2 security, if required.

   Select **Use Java 2 security to restrict application access to local resources** to enforce Java 2 security permission checking.

   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.

   Note: Updates to the app.policy file only apply to the enterprise applications on the node to which the app.policy file belongs.

a. Optional: Select **Warn if applications are granted custom permissions**. The filter.policy file contains a list of permissions that an application should not have according to the J2EE 1.3 Specification. If an application is installed with a permission specified in this policy file and this option is enabled, a warning is issued. The default is enabled.

b. Optional: Select **Restrict access to resource authentication data**. Enable this option if you need to restrict application access to sensitive Java Connector Architecture (JCA) mapping authentication data.

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.

What to do next

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

Related information

⇨ Configuring Java 2 security policy files

Administrative security

Administrative security determines whether security is used at all, the type of registry against which authentication takes place, and other values, many of which act as defaults. Proper planning is required because incorrectly enabling administrative security can lock you out of the administrative console or cause the server to end abnormally.

Administrative security can be thought of as a ″big switch″ that activates a wide variety of security settings for WebSphere Process Server. Values for these settings can be specified, but they will not take effect until administrative security is activated. The settings include the authentication of users, the use of Secure Sockets Layer (SSL), and the choice of user account repository. In particular, application security, including authentication and role-based authorization, is not enforced unless administrative security is active. Administrative security is enabled by default.

Administrative security represents the security configuration that is effective for the entire security domain. A security domain consists of all of the servers that are configured with the same user registry realm name. In some cases, the realm can be the machine name of a local operating system registry. In this case, all of the application servers must reside on the same physical machine. In other cases, the realm can be the machine name of a standalone Lightweight Directory Access Protocol (LDAP) registry.

A multiple node configuration is supported because you can access remotely user registries that support the LDAP protocol. Therefore, you can enable authentication from anywhere.

The basic requirement for a security domain is that the access ID that is returned by the registry or repository from one server within the security domain is the same access ID as that returned from the registry or repository on any other server within the same security domain. The access ID is the unique identification of a user and is used during authorization to determine if access is permitted to the resource.

The administrative security configuration applies to every server within the security domain.

Why turn on administrative security?

Turning on administrative security activates the settings that protect your server from unauthorized users. Administrative security is enabled by default during profile creation. There might be some environments where no security is needed such as a development system. On these systems you can elect to disable administrative security. However, in most environments you should keep unauthorized users from accessing the administrative console and your business applications. Administrative security must be enabled to restrict access.

What does administrative security protect?

The configuration of administrative security for a security domain involves configuring the following technologies:

- Authentication of HTTP clients
- Authentication of IIOP clients
- Administrative console security
- Naming security
- Use of SSL transports
- Role-based authorization checks of servlets, enterprise beans, and MBeans
- Propagation of identities (RunAs)
- CBIND checks
- The common user registry
- The authentication mechanism
- Other security information that defines the behavior of a security domain includes:
  - The authentication protocol (Remote Method Invocation over the Internet Inter-ORB Protocol (RMI/IIOP) security)
  - Other miscellaneous attributes

Application security

Application security enables security for the applications in your environment. This type of security provides application isolation and requirements for authenticating application users.

In previous releases of WebSphere Process Server, when a user enabled global security, both administrative and application security were enabled. The notion of global security is now split into administrative security and application security, each of which you can enable separately.

Administrative security is enabled, by default. Application security is also enabled by default. Application security is in effect only when administrative security is enabled.

Java 2 security

Java 2 security provides a policy-based, fine-grain access control mechanism that increases overall system integrity by checking for permissions before allowing access to certain protected system resources. Java 2 security guards access to system resources such as file I/O, sockets, and properties. Java 2 Platform, Enterprise Edition (J2EE) security guards access to Web resources such as servlets, JavaServer Pages (JSP) files and Enterprise JavaBeans™ (EJB) methods.

WebSphere Process Server security includes the following technologies:

- Java 2 Security Manager
- Java Authentication and Authorization Service (JAAS)
- Java 2 Connector authentication data entries
- J2EE role-based authorization
- Secure Sockets Layer (SSL) configuration

Because Java 2 security is relatively new, many existing or even new applications might not be prepared for the very fine-grain access control programming model that Java 2 security is capable of enforcing. Administrators need to understand the possible consequences of enabling Java 2 security if applications are not prepared for Java 2 security. Java 2 security places some new requirements on application developers and administrators.

See related information for more details on Java 2 security.

Related information

Java 2 security

Configuring a user account repository

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

About this task

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

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

Procedure

1. Navigate to the Secure administration, applications, and infrastructure panel in the administrative console. Expand **Security** and click on **Secure administration, applications, and infrastructure**.
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
Federated repositories	Specify this setting to manage profiles in multiple repositories under a single realm. The realm can consist of identities in: • The file-based repository that is built into the system, • One or more external repositories, • Both the built-in, file-based repository and in one or more external repositories. **Note:** Only a user with administrator privileges can view the federated repositories configuration. See Managing the realm in a federated repository configuration for more information.
Local Operating System	The default user registry. Under **Available realm Definitions** select **Local operating system**, click **configure**. 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.
Custom user registry	Choose a custom account repository and configure it to your needs.
Tivoli® Access Manager	**Note:** This option is not available through the administrative console, and must be configured using the wsadmin command.

Configuring the user account repository

You can configure your user account repository using the administrative console. You can choose a server user identity or automatically generate a server identity.

About this task

You can configure the user account repository using the administrative console. You can choose to allow WebSphere Process Server to automatically generate a server user identity or you can specify one from the user account repository that you are employing. The latter choice improves auditability of administrative actions.

When using LDAP as the user registry for WebSphere Process Server for z/OS, security is administered using the administrative console. If you use the operating system for the user registry, authorize security using the System Authorization Facility.

Procedure

1. From the administrative console open the **User account repository** configuration page for your user registry.

   Expand **Security**, click **Secure administration, applications, and infrastructure**, and select the user registry that you are employing under the **Available realm definitions** menu. Click **Configure**.
2. Optional: Enter a **Primary administrative user name**. Specifies the name of a user with administrative privileges that is defined in your local operating system. The user name is used to log on to the administrative console when administrative security is enabled.
3. Select either the **Automatically generated server identity** or **Server identity that is stored in the repository** option.

   If you select the **Server identity that is stored in the repository** option, enter the following information:
   - Server user ID or administrative user.
   - Associated password for this user.

   This identity must exist in the user account repository.

Configuring WebSphere Process Server to use Tivoli Access Manager as the user account repository

You can use Tivoli Access Manager as your user account repository, you must configure it using the wsadmin command, outside of the administrative console.

About this task

The Tivoli Access Manager can be used as the user account repository. You cannot configure it on the administrative console and must use the wsadmin command. See the WebSphere Application Server Information Center topic: Propagating security policy of installed applications to a JACC provider using wsadmin scripting.

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.

About this task

This task assumes that you have global security turned on.

Procedure

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

   Expand **Security**, click **Secure administration, applications, and infrastructure**, and select **LDAP** under the **Available realm definitions** menu. Click **Configure**.
4. Enter a valid user name in the **Primary administrative user name** field. This value is the name of a user with administrative privileges that is defined in the registry. This user name is used to access the administrative console or used by the wsadmin command.
5. Click **Apply**.

6. Select either the **Automatically generated server identity** or **Server identity that is stored in the repository** option.

   If you select the **Server identity that is stored in the repository** option, enter the following information:

   - Server user ID or administrative user.
   - Associated password for this user.

   Although this ID is not the LDAP administrator user ID, however, the entry must exist in the LDAP.
7. 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.
8. Enter the name of the computer where the LDAP resides.

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

   In the **Port** field, enter the port number on which the LDAP server is listening.
10. 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.
11. Leave the remaining parameters with the default values and confirm your changes.

    Click **OK**.

Starting and stopping the server

When administrative 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.

Before you begin

Administrative security must be enabled.

Procedure

1. Start the server. At the command prompt in the */AppServer*//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 error message is returned.
2. Stop the server. At the command prompt in the */AppServer*//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

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

There are seven roles provided as part of the administrative console. These roles grant permission to ranges of functionality on the administrative console. When administrative 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 1. 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 .
Adminsecuritymanager	Only users who are granted this role can map users to administrative roles. Also, when fine-grained administrative security is used, only users who are granted this role can manage authorization groups. See Administrative roles for more information.
Deployer	Users who are granted this role can perform both configuration actions and runtime operations on applications.

Table 1. Administrative security roles (continued)

Administrative security role	Description
iscadmins	This role is only available for administrative console users and not for wsadmin users. Users who are granted this role have administrator privileges for managing users and groups in the federated repositories. For example, a user of the iscadmins role can complete the following tasks: • Create, update, or delete users in the federated repositories configuration. • Create, update, or delete groups in the federated repositories configuration.

The server ID that is specified when you enable administrative 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.

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 predefined 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 2 on page 16 are used to invoke the components regardless of the identity of the invoking user.

Table 2. Authentication aliases associated with business processes.

Alias	Description	Information
BPEAuthDataAliasJMS_*node_server*	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*	Used to authenticate with databases.	Configure the database using the provided scripts.

Table 3 describes the RunAs roles created for business processes.

Table 3. 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.
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 4 are used to invoke the components regardless of the identity of the invoking user.

Table 4. 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 5 are used to invoke the components regardless of the identity of the invoking user.

Table 5. 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.

Roles	Default permission
eventCreator	All authenticated users.
catalogAdministrator	All authenticated users.
catalogReader	All authenticated users.

Configuring WebSphere Process Server security for a deployment environment server

Follow the links below for how to configure the security of a deployment environment installation of WebSphere Process Server.

Securing a deployment environment of WebSphere Process Server

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.

Before you begin

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

About this task

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.

Procedure

1. Ensure that administrative security is turned on. "Enabling administrative security" on page 7.
2. Ensure that application security is turned on. "Securing applications in WebSphere Process Server" on page 31.
3. Add users or groups to the administrative role. You can give administrative rights to individual users or to a group of users by following the **Administrative User Roles** or **Administrative Group Roles**, respectively.
4. Select the user account repository that 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
Federated repositories	Specify this setting to manage profiles in multiple repositories under a single realm. The realm can consist of identities in: • The file-based repository that is built into the system, • One or more external repositories, • Both the built-in, file-based repository and in one or more external repositories. **Note:** Only a user with administrator privileges can view the federated repositories configuration. See Managing the realm in a federated repository configuration for more information.
Local Operating System	The default user registry. See "Configuring the user account repository" on page 11 for details of how to configure the user account registry.
Standalone LDAP registry	Follow the instructions in Configuring Lightweight Directory Access Protocol (LDAP) as the user registry to configure LDAP as your user registry.
Standalone custom registry	See "Configuring the user account repository" on page 11 for details of how to configure the user account registry.

5. Apply these changes.

   Click the **Apply** button at the bottom of the panel.
6. Go to the Business Integration Security panel. Expand **Security** and click **Business Integration Security**.
7. Supply appropriate user identities for the listed authentication aliases. The credential you provide must exist in the user account repository that you are employing. It is important for the security of your system that you choose appropriate user identities to act as authentication aliases.
8. On the same panel you can configure security for Business Process Choreographer.

   Set the business process choreographer user role mappings for the business flow and human task manager:

   - **Administrator**: User name(s) and/or group name(s) for the business flow and human task administrator role. Users assigned to this role have all privileges.
   - **Monitor**: User name(s) and/or group name(s) for the business flow and human task monitor role. Users assigned to this role can view the properties of all of the business process and task objects.

   The business process choreographer authentication aliases can be configured for each deployment target where the business process choreographer has been installed. The following authentication aliases are listed:

   - **JMS API Authentication**: Authentication for the business flow manager message-driven bean to process asynchronous API calls.
   - **Escalation User Authentication**: Authentication for the human task manager message-driven bean to process asynchronous API calls.
9. Apply these changes.

Click the **Apply** button at the bottom of the panel.

10. Save the changes to the local configuration.

    Click **Save** in the message pane.
11. Ensure that the security information is propagated to the nodes of the cell.

    Expand **System administration** on the administrative console and click **Nodes**. Click **Full Resynchronize**.
12. 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.

Results

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. The system administrator user identity may have been used in multiple places during installation and configuration of environment. It is advisable to replace this identity with appropriate user credentials from the user account repository for all but the core security functions. Use the **Business Integration Security** panel in the administrative console to administer these identities and aliases.

Enabling administrative security

In WebSphere Application Server version 6.1, administrative security is enabled by default. If you have disabled administrative security, the following instructions describe how to enable it again.

Before you begin

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

About this task

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.

Procedure

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

   Expand **Security** and click **Secure administration, applications, and infrastructure**.
2. Enable administrative security.

   Select **Enable administrative security**.
3. Optional: Enforce Java 2 security, if required.

   Select **Use Java 2 security to restrict application access to local resources** to enforce Java 2 security permission checking.

   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.

Note: Updates to the app.policy file only apply to the enterprise applications on the node to which the app.policy file belongs.

a. Optional: Select **Warn if applications are granted custom permissions**. The filter.policy file contains a list of permissions that an application should not have according to the J2EE 1.3 Specification. If an application is installed with a permission specified in this policy file and this option is enabled, a warning is issued. The default is enabled.
b. Optional: Select **Restrict access to resource authentication data**. Enable this option if you need to restrict application access to sensitive Java Connector Architecture (JCA) mapping authentication data.

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.

What to do next

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

Related information

Configuring Java 2 security policy files

Administrative security

Administrative security determines whether security is used at all, the type of registry against which authentication takes place, and other values, many of which act as defaults. Proper planning is required because incorrectly enabling administrative security can lock you out of the administrative console or cause the server to end abnormally.

Administrative security can be thought of as a ″big switch″ that activates a wide variety of security settings for WebSphere Process Server. Values for these settings can be specified, but they will not take effect until administrative security is activated. The settings include the authentication of users, the use of Secure Sockets Layer (SSL), and the choice of user account repository. In particular, application security, including authentication and role-based authorization, is not enforced unless administrative security is active. Administrative security is enabled by default.

Administrative security represents the security configuration that is effective for the entire security domain. A security domain consists of all of the servers that are configured with the same user registry realm name. In some cases, the realm can be the machine name of a local operating system registry. In this case, all of the application servers must reside on the same physical machine. In other cases, the realm can be the machine name of a standalone Lightweight Directory Access Protocol (LDAP) registry.

A multiple node configuration is supported because you can access remotely user registries that support the LDAP protocol. Therefore, you can enable authentication from anywhere.

The basic requirement for a security domain is that the access ID that is returned by the registry or repository from one server within the security domain is the same access ID as that returned from the registry or repository on any other server within the same security domain. The access ID is the unique identification of a user and is used during authorization to determine if access is permitted to the resource.

The administrative security configuration applies to every server within the security domain.

Why turn on administrative security?

Turning on administrative security activates the settings that protect your server from unauthorized users. Administrative security is enabled by default during profile creation. There might be some environments where no security is needed such as a development system. On these systems you can elect to disable administrative security. However, in most environments you should keep unauthorized users from accessing the administrative console and your business applications. Administrative security must be enabled to restrict access.

What does administrative security protect?

The configuration of administrative security for a security domain involves configuring the following technologies:

- Authentication of HTTP clients
- Authentication of IIOP clients
- Administrative console security
- Naming security
- Use of SSL transports
- Role-based authorization checks of servlets, enterprise beans, and MBeans
- Propagation of identities (RunAs)
- CBIND checks
- The common user registry
- The authentication mechanism
- Other security information that defines the behavior of a security domain includes:
  - The authentication protocol (Remote Method Invocation over the Internet Inter-ORB Protocol (RMI/IIOP) security)
  - Other miscellaneous attributes

Application security

Application security enables security for the applications in your environment. This type of security provides application isolation and requirements for authenticating application users.

In previous releases of WebSphere Process Server, when a user enabled global security, both administrative and application security were enabled. The notion of global security is now split into administrative security and application security, each of which you can enable separately.

Administrative security is enabled, by default. Application security is also enabled by default. Application security is in effect only when administrative security is enabled.

Java 2 security

Java 2 security provides a policy-based, fine-grain access control mechanism that increases overall system integrity by checking for permissions before allowing access to certain protected system resources. Java 2 security guards access to system resources such as file I/O, sockets, and properties. Java 2 Platform, Enterprise Edition (J2EE) security guards access to Web resources such as servlets, JavaServer Pages (JSP) files and Enterprise JavaBeans (EJB) methods.

WebSphere Process Server security includes the following technologies:

- Java 2 Security Manager
- Java Authentication and Authorization Service (JAAS)
- Java 2 Connector authentication data entries
- J2EE role-based authorization
- Secure Sockets Layer (SSL) configuration

Because Java 2 security is relatively new, many existing or even new applications might not be prepared for the very fine-grain access control programming model that Java 2 security is capable of enforcing. Administrators need to understand the possible consequences of enabling Java 2 security if applications are not prepared for Java 2 security. Java 2 security places some new requirements on application developers and administrators.

See related information for more details on Java 2 security.

Related information

Java 2 security

Configuring a user account repository

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

About this task

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

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

Procedure

1. Navigate to the Secure administration, applications, and infrastructure panel in the administrative console. Expand **Security** and click on **Secure administration, applications, and infrastructure**.
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
Federated repositories	Specify this setting to manage profiles in multiple repositories under a single realm. The realm can consist of identities in: • The file-based repository that is built into the system, • One or more external repositories, • Both the built-in, file-based repository and in one or more external repositories. **Note:** Only a user with administrator privileges can view the federated repositories configuration. See Managing the realm in a federated repository configuration for more information.
Local Operating System	The default user registry. Under **Available realm Definitions** select **Local operating system**, click **configure**. 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.
Custom user registry	Choose a custom account repository and configure it to your needs.
Tivoli Access Manager	**Note:** This option is not available through the administrative console, and must be configured using the wsadmin command.

Configuring the user account repository

You can configure your user account repository using the administrative console. You can choose a server user identity or automatically generate a server identity.

About this task

You can configure the user account repository using the administrative console. You can choose to allow WebSphere Process Server to automatically generate a server user identity or you can specify one from the user account repository that you are employing. The latter choice improves auditability of administrative actions.

When using LDAP as the user registry for WebSphere Process Server for z/OS, security is administered using the administrative console. If you use the operating system for the user registry, authorize security using the System Authorization Facility.

Procedure

1. From the administrative console open the **User account repository** configuration page for your user registry.

   Expand **Security**, click **Secure administration, applications, and infrastructure**, and select the user registry that you are employing under the **Available realm definitions** menu. Click **Configure**.
2. Optional: Enter a **Primary administrative user name**. Specifies the name of a user with administrative privileges that is defined in your local operating system. The user name is used to log on to the administrative console when administrative security is enabled.
3. Select either the **Automatically generated server identity** or **Server identity that is stored in the repository** option.

   If you select the **Server identity that is stored in the repository** option, enter the following information:
   - Server user ID or administrative user.
   - Associated password for this user.

   This identity must exist in the user account repository.

Configuring WebSphere Process Server to use Tivoli Access Manager as the user account repository

You can use Tivoli Access Manager as your user account repository, you must configure it using the wsadmin command, outside of the administrative console.

About this task

The Tivoli Access Manager can be used as the user account repository. You cannot configure it on the administrative console and must use the wsadmin command. See the WebSphere Application Server Information Center topic: Propagating security policy of installed applications to a JACC provider using wsadmin scripting.

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.

About this task

This task assumes that you have global security turned on.

Procedure

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

   Expand **Security**, click **Secure administration, applications, and infrastructure**, and select **LDAP** under the **Available realm definitions** menu. Click **Configure**.
4. Enter a valid user name in the **Primary administrative user name** field. This value is the name of a user with administrative privileges that is defined in the registry. This user name is used to access the administrative console or used by the wsadmin command.
5. Click **Apply**.

6. Select either the **Automatically generated server identity** or **Server identity that is stored in the repository** option.

   If you select the **Server identity that is stored in the repository** option, enter the following information:

   - Server user ID or administrative user.
   - Associated password for this user.

   Although this ID is not the LDAP administrator user ID, however, the entry must exist in the LDAP.
7. 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.
8. Enter the name of the computer where the LDAP resides.

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

   In the **Port** field, enter the port number on which the LDAP server is listening.
10. 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.
11. Leave the remaining parameters with the default values and confirm your changes.

    Click **OK**.

Starting and stopping the server

When administrative 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.

Before you begin

Administrative security must be enabled.

Procedure

1. Start the server. At the command prompt in the */AppServer/*/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 error message is returned.
2. Stop the server. At the command prompt in the */AppServer/*/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

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

There are seven roles provided as part of the administrative console. These roles grant permission to ranges of functionality on the administrative console. When administrative 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 6. 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 .
Adminsecuritymanager	Only users who are granted this role can map users to administrative roles. Also, when fine-grained administrative security is used, only users who are granted this role can manage authorization groups. See Administrative roles for more information.
Deployer	Users who are granted this role can perform both configuration actions and runtime operations on applications.

Table 6. Administrative security roles (continued)

Administrative security role	Description
iscadmins	This role is only available for administrative console users and not for wsadmin users. Users who are granted this role have administrator privileges for managing users and groups in the federated repositories. For example, a user of the iscadmins role can complete the following tasks: • Create, update, or delete users in the federated repositories configuration. • Create, update, or delete groups in the federated repositories configuration.

The server ID that is specified when you enable administrative 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.

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 predefined 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 2 on page 16 are used to invoke the components regardless of the identity of the invoking user.

Table 7. Authentication aliases associated with business processes.

Alias	Description	Information
BPEAuthDataAliasJMS_*node_server*	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*	Used to authenticate with databases.	Configure the database using the provided scripts.

Table 3 on page 16 describes the RunAs roles created for business processes.

Table 8. 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.
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 4 on page 16 are used to invoke the components regardless of the identity of the invoking user.

Table 9. 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 5 on page 17 are used to invoke the components regardless of the identity of the invoking user.

Table 10. 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.

Roles	Default permission
eventCreator	All authenticated users.
catalogAdministrator	All authenticated users.
catalogReader	All authenticated users.

Securing applications in WebSphere Process Server

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

Before you begin

Securing your applications assumes that you have administrative security enabled.

About this task

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:

Procedure

1. Ensure that administrative security is enabled. See "Enabling administrative security" on page 7 for more details.
2. Ensure that application security is enabled. On the administrative console, expand **Security** and click **Secure administration, applications, and infrastructure**. Select **Enable application security** in order that WebSphere Process Server will require authentication from users who try to access a secured application.
3. Develop your applications in WebSphere Process Server using all appropriate security features.
4. Deploy your applications to your WebSphere Process Server environment assigning users or groups to appropriate security roles.
5. Maintain the security of your WebSphere Process Server environment.

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 administrative 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 11 lists typical clients that would invoke WebSphere Process Server components, and the authentication options available for each type of client.

Table 11. 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.

About this task

Modify authentication aliases from the administrative console.

Procedure

1. Access the Business Integration Authentication Alias panel.

   From the administrative console, expand **Security**, and click **Business Integration Authentication Alias**.

   Note: You can also access this panel from various administrative console panels which require authentication alias information.
2. Select the authentication alias that you want to modify.

   The Business Integration Authentication Alias panel contains a list of authentication aliases, the associated component, the user ID associated with this alias, and optionally a description of the alias. Click on the alias that you want to modify. Alternatively you can select the check box in the **Select** column corresponding to the authentication alias that you want to edit, and then click the **Edit** button. The authentication alias configuration panel is displayed.
3. Change the properties of the alias.

   On the authentication alias 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 Business Integration Authentication Alias panel, and click **Apply** 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*

   Related tasks

   "Augmenting WebSphere Process Server profiles with security" on page 4

   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.

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. For a more detailed discussion, see related information.

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). WS-Security deployment descriptors can be set in your applications before deployment. See related information for more details.

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.

Related information

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/csec_plan.html http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.wbit.610.help.runtime.doc/topics/tusergoal.html

Configuring a web services web client to use SSL:

You can configure a web services client to invoke a web service using Secure Sockets Layer (SSL).

About this task

The details of how to configure your web services web client to use SSL are provided in this WebSphere Application Server technote. A more general discussion of securing web services can be found in the WebSphere Application Server topic Securing Web services applications at the transport level.

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

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.

Before you begin

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

When developing components for WebSphere Process Server for z/OS, remember that security might be administered using platform-specific facilities, such as the System Authorization Facility.

About this task

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

Procedure

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 begin

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.

About this 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.

Procedure

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.

What to do next

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.

Before you begin

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

About this task

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.

Procedure

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
Security Permission	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 Identity	Provide a valid user name and password for the identity to which the role is mapped.

Related information

Delegations

Securing adapters

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

About this task

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 adapters 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 human tasks and business 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.

Related concepts

Authorization and people assignment for processes

Related information

Authorization and people assignment

Tutorials

Tutorials are provided to guide you through some important security scenarios..

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.

Before you begin

These scenarios all assume that global security is enforced.

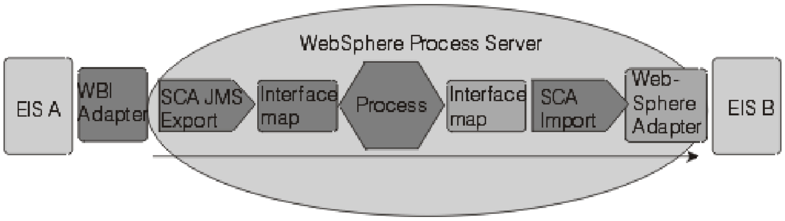
About this task

Procedure

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

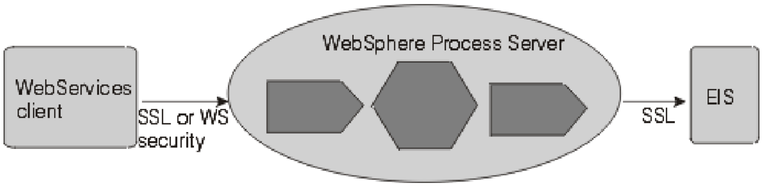
An inbound request comes in from a WebSphere Business Integration Adapter. The service component architecture (SCA) invokes an interface map based on the SCA export. The request flows through a process component, a second interface map and is then passed on to a second EIS (B), via a WebSphere Adapter.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

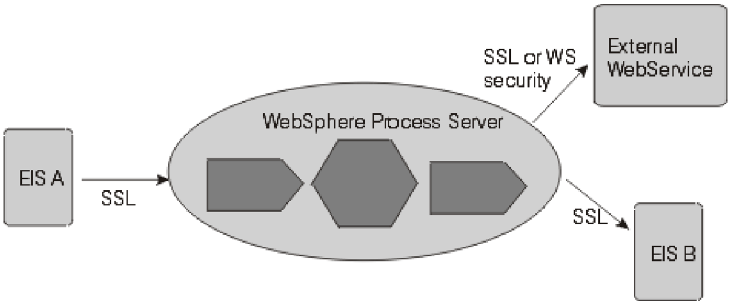
In this scenario a Web service client invokes a WebSphere Process Server component. The request passes through several components in the WebSphere Process Server environment before being passed to an EIS by an adapter.



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.



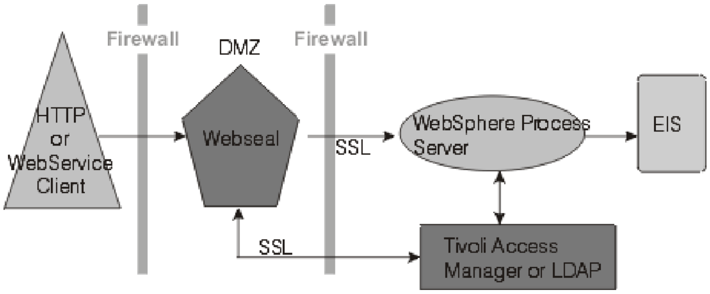
As for the inbound Web service request, you can authenticate with the external Web service as a SSL client, using HTTP Basic authentication or using WS-Security authentication. Use LTPACallBackHandler as the callback mechanism to extract the usernameToken from the current RunAs subject. Between WebSphere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

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 sample Jacl script is located in and . The name of the script is: `RelServicesAdmin.jacl`.

To run the script, enter:script, enter:

```
wsadmin -f install_root/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

About this task

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.

Procedure

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"
       }
   ```

Results

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 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

IBM, the IBM logo, Domino, Tivoli, WebSphere, and z/OS are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

Windows is a registered trademark 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 registered 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 6.1.0

Printed in USA