

WebSphere Process Server for z/OS



Securing Applications and Their Environments

Version 7.0.0

30 April 2010

This edition applies to version 7, release 0, 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.

© **Copyright IBM Corporation 2006, 2010.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Securing WebSphere Process Server and applications 1

General overview of security	1
Getting started with security	2
Installing WebSphere Process Server: security considerations	2
Authentication information provided at installation time	3
Configuring WebSphere Process Server security for a stand-alone server	4
Securing a stand-alone WebSphere Process Server installation	4
Enabling security	5
Configuring a user account repository	7
Starting and stopping the server	12
Administrative security roles	13
Configuring WebSphere Process Server security for a deployment environment server	15

Securing a deployment environment of WebSphere Process Server	15
Enabling security	16
Configuring a user account repository	17
Starting and stopping the server	23
Administrative security roles	24
Securing applications in WebSphere Process Server	26
Elements of application security	26
Deploying (installing) secure applications	34
Security for the Business Calendars widget	38
Securing adapters	41
Setting up security for Business Space	42
Setting application security for Business Space.	43
Configuring Tivoli Access Manager WebSEAL to work with Business Space	47
Assigning the Business Space superuser role	54
Creating end-to-end security.	56

Securing WebSphere Process Server and applications

Security of WebSphere® Process Server and applications depends on securing the runtime environment and securing applications.

Securing the WebSphere Process Server runtime environment involves enabling administrative security, enabling application security, creating profiles with security, and restricting access to critical functions to selected users.

Securing an application includes authenticating users, implementing access control for operations and resources, and providing data integrity and privacy.

WebSphere Process Server security is based on the WebSphere Application Server version 7.0 security. These documents are supplemental to the core security documentation located in the WebSphere Application Server Information Center (specifically the topics in “Securing applications and their environment”).

General overview of security

WebSphere Process Server security is based on the WebSphere Application Server version 7.0 security.

Refer to the WebSphere Application Server Network Deployment 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 can include:

- Authentication of users - A user or a process that invokes an application must be authenticated. With a single sign on, a user can provide authentication data once and then pass this authentication information to downstream components.
- Access control - Does the authenticated user have permission to perform the operation?
- Data integrity and privacy - The data that is accessed by an application must be secured so that no unauthorized party can view or modify it in any way.

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 7.0 security. Considerations that are specific to WebSphere Process Server are listed.

- The administrative console page Business Integration Security is unique to WebSphere Process Server. You display this page by expanding **Security** and clicking **Business Integration Security**.

This page allows users to assign specific identities from their user registry to important business integration authentication aliases. In addition, you can administer your Business Process Choreographer security settings on this page.

- Application security is turned on by default in WebSphere Process Server. This is not the case in WebSphere Application Server.
- WebSphere Process Server contains a set of component-specific security roles.

Getting started with security

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

The following list provides an overview of the tasks you perform when securing WebSphere Process Server.

1. Consider security when you install WebSphere Process Server.
 - a. Secure your environment before installation.
 - b. Prepare the operating system for installation of WebSphere Process Server.
 - c. Prepare your environment after installation.
2. Ensure that security is turned on for your stand-alone or deployment environment installation.
 - a. Ensure that Administrative security is turned on.
 - b. Ensure that Application security is turned on.
 - c. If required, turn on Java™ 2 security.
 - 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. Set up security for specific WebSphere Process Server components. For example, use the Security Roles widget to set up role-based access control for timetables in the Business Calendars widget.
4. 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.
5. Maintain the security of your WebSphere Process Server environment.

Installing WebSphere Process Server: security considerations

Consider how security will be implemented before, during, and after installing WebSphere Process Server.

Procedure

1. Secure your environment before installation.

The commands required to install WebSphere Process Server with proper security 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 Information Center.

2. Prepare the operating system for installation of WebSphere Process Server.

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

3. Secure your environment after installation.

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

What to do next

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

Authentication information provided at installation time

During installation, all 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 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 use 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.

Configuring WebSphere Process Server security for a stand-alone server

Configuring the security of a stand-alone installation of WebSphere Process Server includes such tasks as enabling administrative security and configuring a user account registry.

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.

About this task

The following steps provide a roadmap of the tasks you perform to enable security. More specific details on these tasks are provided in the topics that follow.

Procedure

1. Ensure that administrative security is turned on. "Enabling security."
2. Ensure that application security is turned on. "Enabling security."
3. Select the user account repository that you want to use. "Configuring a user account repository" on page 7
Make sure you set the selected registry as your current registry using **Set as current**.
4. Add users or groups to the administrative role.
5. If necessary, stop and restart the server. "Starting and stopping the server" on page 12
6. Set up authentication aliases, access control, and other security mechanisms for your installed components. "Securing applications in WebSphere Process Server" on page 26

Enabling security

In WebSphere Application Server version 7.0, administrative security is enabled by default. If you have disabled administrative security, use the following instructions to enable it again.

Before you begin

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

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

About this task

With the administrative console, you can enable administrative security, application security, and Java 2 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.

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

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

The administrative security of WebSphere Process Server 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* provides a policy-based, fine-grained 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. It also guards access to Web resources such as servlets, JavaServer Pages (JSP) files, and Enterprise JavaBeans™ (EJB) methods.

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

Attention: Fix packs that include updates to the Software Development Kit (SDK) might overwrite unrestricted policy files. Back up unrestricted policy files before you apply a fix pack and reapply these files after the fix pack is applied.

Procedure

1. Open the administrative security page in the administrative console.
Expand **Security** and click **Global security**.
2. Enable administrative security.
Select **Enable administrative security**.
3. Enable application security.
Select **Enable application security**.
4. 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. Access Control 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 apply only 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 1.4 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.
5. Apply these changes.
Click the **Apply** button at the bottom of the page.
6. Save the changes to the local configuration.
Click **Save** in the message pane.

- 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.

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

- Navigate to the Secure administration, applications, and infrastructure panel in the administrative console. Expand **Security** and click **Global security**.
- 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: <ul style="list-style-type: none"> 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	This is the default user registry. Follow the instructions in “Configuring the local operating system or standalone custom user account repository” on page 8.
Lightweight Directory Access Protocol (LDAP)	Follow the instructions in “Configuring Lightweight Directory Access Protocol (LDAP) as the user registry” on page 9 to configure LDAP as your user registry.

User registry	Action
Custom user registry	Follow the instructions in “Configuring the local operating system or standalone custom user account repository” to choose a custom account repository and configure it to your needs.
Tivoli® Access Manager	Note: This option is not available through the administrative console. It must be configured using the wsadmin command.

Configuring the local operating system or standalone custom user account repository

You can configure your user account repository using the administrative console. The steps for configuring the local operating system, which is the default, or a stand-alone custom user account registry are similar.

About this task

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

- From the administrative console, open the configuration page for your user registry.
Expand **Security**, click **Global security**, and select the user registry that you are employing under the **Available realm definitions** menu. Click **Configure**.
- Optional: 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. It is also used by the wsadmin command.
- Select either the **Automatically generated server identity** or **Server identity that is stored in the repository** option.
 - If you select **Automatically generated server identity**, the application server generates the server identity that is used for internal process communication. You can change this server identity on the Authentication mechanisms and expiration page. To access the Authentication mechanisms and expiration page, click **Security** → **Global security** → **Authentication mechanisms and expiration**. Change the value of the **Internal server ID** field.
 - If you select the **Server identity that is stored in the repository** option, enter the following information:
 - For **Server user ID or administrative user on a Version 7.0 node**, specify a user ID that is used to run the application server for security purposes.
 - For **Password**, specify the password associated with this user.
- Optional: For stand-alone custom registries only, perform the following steps:

- a. Verify that the value in the **Custom registry class name** is correct, or change it if necessary.
 - b. Select or remove the check from **Ignore case for authentication**.
When you select this option, the authorization check is case insensitive.
5. Click **Apply**.
 6. From the bottom of the page, click **Set as current**.
 7. Click **OK** and either **Apply** or **Save**.

What to do next

Save, stop, and restart all servers so that the updates can take effect.

If the server starts without any problems, the setup is correct.

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; however, 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, you can use an external Lightweight Directory Access Protocol (LDAP) as the user registry.

Before you begin

This task assumes that you have administrative security turned on.

To access a user registry using LDAP, you must know a valid user name (ID) and password, the server host and port of the registry server, the base distinguished name (DN) and, if necessary, the bind DN and the bind password.

You can choose any valid user in the user registry that is searchable. You can use any user ID that has the administrative role to log in.

Important: If you configure WebSphere Process Server to use more than one LDAP server, be aware that the value shown in the **Global security** → **Standalone LDAP Registry** page of the administrative console might not reflect the LDAP server that is being used.

Procedure

1. Start the administrative console.
 - If security is currently disabled, you are prompted for a user ID. Log in with any user ID.

- If security is currently enabled, you are prompted for both a user ID and a password. Log in with a predefined administrative user ID and password.
2. Expand **Security** and click **Global security**.
 3. From the Secure administration, applications, and infrastructure page, perform the following steps:
 - a. Make sure **Enable administrative security** is selected.
 - b. From the **Available realm definitions** list, select **Standalone LDAP registry**.
 - c. Click **Configure**.
 4. On the **Configuration** tab of the Standalone LDAP registry page, perform the following steps:
 - a. 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. It is also used by the wsadmin command.
 You can either enter the complete distinguished name (DN) of the user or the short name of the user, as defined by the user filter in the Advanced LDAP settings page.
 - b. Optional: Select either the **Automatically generated server identity** or **Server identity that is stored in the repository** option.
 - If you select **Automatically generated server identity**, the application server generates the server identity that is used for internal process communication.
 You can change this server identity on the Authentication mechanisms and expiration page. To access the Authentication mechanisms and expiration page click **Security** → **Global security** → **Authentication mechanisms and expiration**. Change the value of the **Internal server ID** field.
 - If you select the **Server identity that is stored in the repository** option, enter the following information:
 - For **Server user ID or administrative user on a Version 7.0 node**, specify a user ID that is used to run the application server for security purposes.
 - For **Password**, specify the password associated with this user.
 Although this ID is not the LDAP administrator user ID, the entry must exist in the LDAP.
 - c. Optional: Select the LDAP server to use from the **Type of LDAP server** list.
 The type of LDAP server determines the default filters that are used by WebSphere Process Server. These default filters change the **Type of LDAP server** field to **Custom**, which indicates that custom filters are used. This action occurs after you click **OK** or **Apply** in the Advanced LDAP settings page. Select the **Custom** type from the list and modify the user and group filters to use other LDAP servers, if required.
 IBM Tivoli Directory Server users can select **IBM Tivoli Directory Server** as the directory type. Use the IBM Tivoli Directory Server directory type for better performance.
 - d. In the **Host** field, enter the fully qualified name of the computer where the LDAP resides.
 You can enter either the IP address or domain name system (DNS) name.
 - e. Optional: In the **Port** field, enter the port number on which the LDAP server is listening.

The host name and the port number represent the realm for this LDAP server in the WebSphere Process Server cell. So, if servers in different cells are communicating with each other using Lightweight Third Party Authentication (LTPA) tokens, these realms must match exactly in all the cells.

The default value is 389.

If multiple WebSphere Process Server are installed and configured to run in the same single sign-on domain, or if the WebSphere Process Server interoperates with a previous version of the WebSphere Process Server, make sure that the port number match all configurations.

- f. Optional: Enter the base distinguished name in the **Base Distinguished Name (DN)** field.

The base distinguished name indicates the starting point for LDAP searches in this LDAP directory server. For example, for a user with a DN of cn=John Doe, ou=Rochester, o=IBM, c=US, specify the base DN as any of the following options (assuming a suffix of c=us): ou=Rochester, o=IBM, c=us or o=IBM c=us or c=us.

For authorization purposes, this field is case-sensitive. This specification implies that if a token is received (for example, from another cell or a Lotus Domino[®] server), the base distinguished name (DN) in the server must match exactly the base DN from the other cell or Domino server. If case sensitivity is not a consideration for authorization, enable **Ignore case for authorization**.

In WebSphere Process Server, the distinguished name is normalized according to the Lightweight Directory Access Protocol (LDAP) specification. Normalization consists of removing spaces in the base distinguished name before or after commas and equal symbols. An example of a non-normalized base distinguished name is o = ibm, c = us or o=ibm, c=us. An example of a normalized base distinguished name is o=ibm,c=us.

This field is required for all LDAP directories except for the Domino Directory, where this field is optional.

- g. Optional: Enter the bind DN name in the **Bind distinguished name** field.

The bind DN is required if anonymous binds are not possible on the LDAP server to obtain user and group information.

If the LDAP server is set up to use anonymous binds, leave this field blank. If a name is not specified, the application server binds anonymously. See the Base Distinguished Name field description for examples of distinguished names.

- h. Optional: Enter the password corresponding to the bind DN in the **Bind password** field.

- i. Optional: Modify the **Search time out** value.

This timeout value is the maximum amount of time that the LDAP server waits to send a response to the product client before stopping the request. The default is 120 seconds.

- j. Ensure that **Reuse connection** is selected.

This option specifies that the server should reuse the LDAP connection. Clear this option only in rare situations where a router is used to send requests to multiple LDAP servers and when the router does not support affinity. Leave this option selected for all other situations.

- k. Optional: Verify that **Ignore case for authorization** is enabled.

When you enable this option, the authorization check is case insensitive.

Normally, an authorization check involves checking the complete DN of a user, which is unique in the LDAP server and is case-sensitive. However, when you use either the IBM Directory Server or the Sun ONE (formerly iPlanet) Directory Server LDAP servers, you must enable this option because the group information that is obtained from the LDAP servers is not consistent in case. This inconsistency affects the authorization check only. Otherwise, this field is optional and can be enabled when a case-sensitive authorization check is required.

For example, you might select this option when you use certificates and the certificate contents do not match the case of the entry in the LDAP server. You can also enable **Ignore case for authorization** when you are using single sign-on (SSO) between the product and Lotus Domino.

The default is enabled.

- l. Optional: Select **SSL enabled** if you want to use Secure Sockets Layer communications with the LDAP server.

If you select the **SSL enabled** option, you can select either **Centrally managed** or **Use specific SSL alias**.

- **Centrally managed**

This option enables you to specify an SSL configuration for a particular scope such as the cell, node, server, or cluster in one location. To use the **Centrally managed** option, you must specify the SSL configuration for the particular set of endpoints.

The Manage endpoint security configurations page displays all the inbound and outbound endpoints that use the SSL protocol.

Expand the **Inbound** or **Outbound** section of the Manage endpoint security configurations page and click the name of a node to specify an SSL configuration that is used for every endpoint on that node. For an LDAP registry, you can override the inherited SSL configuration by specifying an SSL configuration for LDAP.

- **Use specific SSL alias**

This option is used to select one of the SSL configurations in the list below the option.

This configuration is used only when SSL is enabled for LDAP. The default is **NodeDefaultSSLSettings**.

- m. Click **OK** and either **Apply** or **Save** until you return to the Secure administration, applications, and infrastructure page.
5. From the Secure administration, applications, and infrastructure page, click **Set as current**.
6. Click **OK** and either **Apply** or **Save**.

What to do next

Save, stop, and restart all servers so that the updates can take effect.

If the server starts without any problems, the setup is correct.

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.

Avoid trouble: **Vista** **Windows 7** If User Account Control (UAC) is enabled on some levels, the application server must be started with Administrator privileges if you are using a command prompt. Start the application server from a command prompt window that is launched by performing the following actions:

- Right-click a command prompt shortcut.
- Click **Run As Administrator**.
- When you open the command prompt window as Administrator, an operating-system dialog appears that asks you if you want to continue. Click **Continue** to proceed.

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 eight 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 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.

Table 1. Administrative security roles (continued)

Administrative security role	Description
Operator	A member of the operator role has monitor privileges, plus the ability to modify the runtime state (that is, start and stop the server).
Administrator	<p>The administrator role is a combination of configurator and operator roles plus additional privileges granted solely to the administrator role. Examples include:</p> <ul style="list-style-type: none"> • Modifying the server user ID and password • Mapping users and groups to the administrator role <p>The administrator also has the permission required to access sensitive information, such as:</p> <ul style="list-style-type: none"> • Lightweight Third Party Authentication (LTPA) passwords • Keys
ISC Admins	<p>This role is available only 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 ISC Admins role can complete the following tasks:</p> <ul style="list-style-type: none"> • Create, update, or delete users in the federated repositories configuration • Create, update, or delete groups in the federated repositories configuration
Deployer	Users who are granted this role can perform both configuration actions and runtime operations on applications.
Admin Security Manager	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.
Auditor	<p>Users who are granted this role can view and modify the configuration settings for the security auditing subsystem.</p> <p>Note: The auditor role includes the monitor role. This allows the auditor to view but not change the rest of the security configuration.</p>

See Administrative roles in the WebSphere Application Server information center for more information.

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.

Tip: 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.

The failed event manager can be operated by any user granted either the administrator or the operator role.

Selectors can be configured by any user granted either the administrator or the configurator role

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 were not enabled.

Configuring WebSphere Process Server security for a deployment environment server

Configuring the security of a deployment environment installation of WebSphere Process Server includes such tasks as enabling administrative security and configuring a user account registry.

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.

About this task

The following steps provide a roadmap of the tasks you perform to enable security. More specific details on these tasks are provided in the topics that follow.

Procedure

1. Ensure that administrative security is turned on. “Enabling security” on page 5.
2. Ensure that application security is turned on. “Enabling security” on page 5.
3. Select the user account repository that you want to use. “Configuring a user account repository” on page 7
Make sure you set the selected registry as your current registry using **Set as current**.
4. Add users or groups to the administrative role.
5. If necessary, stop and restart the server. “Starting and stopping the server” on page 12

6. Set up authentication aliases, access control, and other security mechanisms for your installed components. "Securing applications in WebSphere Process Server" on page 26

Enabling security

In WebSphere Application Server version 7.0, administrative security is enabled by default. If you have disabled administrative security, use the following instructions to enable it again.

Before you begin

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

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

About this task

With the administrative console, you can enable administrative security, application security, and Java 2 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.

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

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

The administrative security of WebSphere Process Server 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* provides a policy-based, fine-grained 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. It also guards access to Web resources such as servlets, JavaServer Pages (JSP) files, and Enterprise JavaBeans (EJB) methods.

Because Java 2 security is relatively new, many existing or even new applications might not be prepared for the very fine-grained access control programming model that it is capable of enforcing. Administrators need to understand the

possible consequences of enabling Java 2 security if applications are not prepared for it. Java 2 Security places some new requirements on application developers and administrators.

Attention: Fix packs that include updates to the Software Development Kit (SDK) might overwrite unrestricted policy files. Back up unrestricted policy files before you apply a fix pack and reapply these files after the fix pack is applied.

Procedure

1. Open the administrative security page in the administrative console.
Expand **Security** and click **Global security**.
2. Enable administrative security.
Select **Enable administrative security**.
3. Enable application security.
Select **Enable application security**.
4. 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. Access Control 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 apply only 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 1.4 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.
5. Apply these changes.
Click the **Apply** button at the bottom of the page.
6. Save the changes to the local configuration.
Click **Save** in the message pane.
7. 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.

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 **Global security**.
2. Select the user registry you want to use.

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

User registry	Action
Federated repositories	Specify this setting to manage profiles in multiple repositories under a single realm. The realm can consist of identities in: <ul style="list-style-type: none"> • 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. <p>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.</p>
Local Operating System	This is the default user registry. Follow the instructions in “Configuring the local operating system or standalone custom user account repository” on page 8.
Lightweight Directory Access Protocol (LDAP)	Follow the instructions in “Configuring Lightweight Directory Access Protocol (LDAP) as the user registry” on page 9 to configure LDAP as your user registry.
Custom user registry	Follow the instructions in “Configuring the local operating system or standalone custom user account repository” on page 8 to choose a custom account repository and configure it to your needs.
Tivoli Access Manager	Note: This option is not available through the administrative console. It must be configured using the wsadmin command.

Configuring the local operating system or standalone custom user account repository

You can configure your user account repository using the administrative console. The steps for configuring the local operating system, which is the default, or a stand-alone custom user account registry are similar.

About this task

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 configuration page for your user registry.
Expand **Security**, click **Global security**, and select the user registry that you are employing under the **Available realm definitions** menu. Click **Configure**.
2. Optional: 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. It is also used by the wsadmin command.
3. Select either the **Automatically generated server identity** or **Server identity that is stored in the repository** option.
 - If you select **Automatically generated server identity**, the application server generates the server identity that is used for internal process communication. You can change this server identity on the Authentication mechanisms and expiration page. To access the Authentication mechanisms and expiration page, click **Security** → **Global security** → **Authentication mechanisms and expiration**. Change the value of the **Internal server ID** field.
 - If you select the **Server identity that is stored in the repository** option, enter the following information:
 - For **Server user ID or administrative user on a Version 7.0 node**, specify a user ID that is used to run the application server for security purposes.
 - For **Password**, specify the password associated with this user.
4. Optional: For stand-alone custom registries only, perform the following steps:
 - a. Verify that the value in the **Custom registry class name** is correct, or change it if necessary.
 - b. Select or remove the check from **Ignore case for authentication**.
When you select this option, the authorization check is case insensitive.
5. Click **Apply**.
6. From the bottom of the page, click **Set as current**.
7. Click **OK** and either **Apply** or **Save**.

What to do next

Save, stop, and restart all servers so that the updates can take effect.

If the server starts without any problems, the setup is correct.

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; however, 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, you can use an external Lightweight Directory Access Protocol (LDAP) as the user registry.

Before you begin

This task assumes that you have administrative security turned on.

To access a user registry using LDAP, you must know a valid user name (ID) and password, the server host and port of the registry server, the base distinguished name (DN) and, if necessary, the bind DN and the bind password.

You can choose any valid user in the user registry that is searchable. You can use any user ID that has the administrative role to log in.

Important: If you configure WebSphere Process Server to use more than one LDAP server, be aware that the value shown in the **Global security** → **Standalone LDAP Registry** page of the administrative console might not reflect the LDAP server that is being used.

Procedure

1. Start the administrative console.
 - If security is currently disabled, you are prompted for a user ID. Log in with any user ID.
 - If security is currently enabled, you are prompted for both a user ID and a password. Log in with a predefined administrative user ID and password.
2. Expand **Security** and click **Global security**.
3. From the Secure administration, applications, and infrastructure page, perform the following steps:
 - a. Make sure **Enable administrative security** is selected.
 - b. From the **Available realm definitions** list, select **Standalone LDAP registry**.
 - c. Click **Configure**.

4. On the **Configuration** tab of the Standalone LDAP registry page, perform the following steps:
 - a. 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. It is also used by the wsadmin command.

You can either enter the complete distinguished name (DN) of the user or the short name of the user, as defined by the user filter in the Advanced LDAP settings page.
 - b. Optional: Select either the **Automatically generated server identity** or **Server identity that is stored in the repository** option.
 - If you select **Automatically generated server identity**, the application server generates the server identity that is used for internal process communication.

You can change this server identity on the Authentication mechanisms and expiration page. To access the Authentication mechanisms and expiration page click **Security** → **Global security** → **Authentication mechanisms and expiration**. Change the value of the **Internal server ID** field.
 - If you select the **Server identity that is stored in the repository** option, enter the following information:
 - For **Server user ID or administrative user on a Version 7.0 node**, specify a user ID that is used to run the application server for security purposes.
 - For **Password**, specify the password associated with this user.

Although this ID is not the LDAP administrator user ID, the entry must exist in the LDAP.
 - c. Optional: Select the LDAP server to use from the **Type of LDAP server** list.

The type of LDAP server determines the default filters that are used by WebSphere Process Server. These default filters change the **Type of LDAP server** field to **Custom**, which indicates that custom filters are used. This action occurs after you click **OK** or **Apply** in the Advanced LDAP settings page. Select the **Custom** type from the list and modify the user and group filters to use other LDAP servers, if required.

IBM Tivoli Directory Server users can select **IBM Tivoli Directory Server** as the directory type. Use the IBM Tivoli Directory Server directory type for better performance.
 - d. In the **Host** field, enter the fully qualified name of the computer where the LDAP resides.

You can enter either the IP address or domain name system (DNS) name.
 - e. Optional: In the **Port** field, enter the port number on which the LDAP server is listening.

The host name and the port number represent the realm for this LDAP server in the WebSphere Process Server cell. So, if servers in different cells are communicating with each other using Lightweight Third Party Authentication (LTPA) tokens, these realms must match exactly in all the cells.

The default value is 389.

If multiple WebSphere Process Server are installed and configured to run in the same single sign-on domain, or if the WebSphere Process Server

interoperates with a previous version of the WebSphere Process Server, make sure that the port number match all configurations.

- f. Optional: Enter the base distinguished name in the **Base Distinguished Name (DN)** field.

The base distinguished name indicates the starting point for LDAP searches in this LDAP directory server. For example, for a user with a DN of cn=John Doe, ou=Rochester, o=IBM, c=US, specify the base DN as any of the following options (assuming a suffix of c=us): ou=Rochester, o=IBM, c=us or o=IBM c=us or c=us.

For authorization purposes, this field is case-sensitive. This specification implies that if a token is received (for example, from another cell or a Lotus Domino server), the base distinguished name (DN) in the server must match exactly the base DN from the other cell or Domino server. If case sensitivity is not a consideration for authorization, enable **Ignore case for authorization**.

In WebSphere Process Server, the distinguished name is normalized according to the Lightweight Directory Access Protocol (LDAP) specification. Normalization consists of removing spaces in the base distinguished name before or after commas and equal symbols. An example of a non-normalized base distinguished name is o = ibm, c = us or o=ibm, c=us. An example of a normalized base distinguished name is o=ibm,c=us.

This field is required for all LDAP directories except for the Domino Directory, where this field is optional.

- g. Optional: Enter the bind DN name in the **Bind distinguished name** field.

The bind DN is required if anonymous binds are not possible on the LDAP server to obtain user and group information.

If the LDAP server is set up to use anonymous binds, leave this field blank. If a name is not specified, the application server binds anonymously. See the Base Distinguished Name field description for examples of distinguished names.

- h. Optional: Enter the password corresponding to the bind DN in the **Bind password** field.

- i. Optional: Modify the **Search time out** value.

This timeout value is the maximum amount of time that the LDAP server waits to send a response to the product client before stopping the request. The default is 120 seconds.

- j. Ensure that **Reuse connection** is selected.

This option specifies that the server should reuse the LDAP connection. Clear this option only in rare situations where a router is used to send requests to multiple LDAP servers and when the router does not support affinity. Leave this option selected for all other situations.

- k. Optional: Verify that **Ignore case for authorization** is enabled.

When you enable this option, the authorization check is case insensitive.

Normally, an authorization check involves checking the complete DN of a user, which is unique in the LDAP server and is case-sensitive. However, when you use either the IBM Directory Server or the Sun ONE (formerly iPlanet) Directory Server LDAP servers, you must enable this option because the group information that is obtained from the LDAP servers is not consistent in case. This inconsistency affects the authorization check only. Otherwise, this field is optional and can be enabled when a case-sensitive authorization check is required.

For example, you might select this option when you use certificates and the certificate contents do not match the case of the entry in the LDAP server. You can also enable **Ignore case for authorization** when you are using single sign-on (SSO) between the product and Lotus Domino.

The default is enabled.

- l. Optional: Select **SSL enabled** if you want to use Secure Sockets Layer communications with the LDAP server.

If you select the **SSL enabled** option, you can select either **Centrally managed** or **Use specific SSL alias**.

- **Centrally managed**

This option enables you to specify an SSL configuration for a particular scope such as the cell, node, server, or cluster in one location. To use the **Centrally managed** option, you must specify the SSL configuration for the particular set of endpoints.

The Manage endpoint security configurations page displays all the inbound and outbound endpoints that use the SSL protocol.

Expand the **Inbound** or **Outbound** section of the Manage endpoint security configurations page and click the name of a node to specify an SSL configuration that is used for every endpoint on that node. For an LDAP registry, you can override the inherited SSL configuration by specifying an SSL configuration for LDAP.

- **Use specific SSL alias**

This option is used to select one of the SSL configurations in the list below the option.

This configuration is used only when SSL is enabled for LDAP. The default is **NodeDefaultSSLSettings**.

- m. Click **OK** and either **Apply** or **Save** until you return to the Secure administration, applications, and infrastructure page.
5. From the Secure administration, applications, and infrastructure page, click **Set as current**.
6. Click **OK** and either **Apply** or **Save**.

What to do next

Save, stop, and restart all servers so that the updates can take effect.



If the server starts without any problems, the setup is correct.

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.

Avoid trouble:   If User Account Control (UAC) is enabled on some levels, the application server must be started with Administrator privileges if you are using a command prompt. Start the application server from a command prompt window that is launched by performing the following actions:

- Right-click a command prompt shortcut.
- Click **Run As Administrator**.
- When you open the command prompt window as Administrator, an operating-system dialog appears that asks you if you want to continue. Click **Continue** to proceed.

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 eight 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 roles in order to access the administrative console.

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

Table 2. Administrative security roles

Administrative security role	Description
Monitor	A member of the monitor role can view the WebSphere Process Server configuration and the current state of the server.
Configurator	A member of the configurator role can edit the WebSphere Process Server configuration.
Operator	A member of the operator role has monitor privileges, plus the ability to modify the runtime state (that is, start and stop the server).

Table 2. Administrative security roles (continued)

Administrative security role	Description
Administrator	<p>The administrator role is a combination of configurator and operator roles plus additional privileges granted solely to the administrator role. Examples include:</p> <ul style="list-style-type: none"> • Modifying the server user ID and password • Mapping users and groups to the administrator role <p>The administrator also has the permission required to access sensitive information, such as:</p> <ul style="list-style-type: none"> • Lightweight Third Party Authentication (LTPA) passwords • Keys
ISC Admins	<p>This role is available only 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 ISC Admins role can complete the following tasks:</p> <ul style="list-style-type: none"> • Create, update, or delete users in the federated repositories configuration • Create, update, or delete groups in the federated repositories configuration
Deployer	<p>Users who are granted this role can perform both configuration actions and runtime operations on applications.</p>
Admin Security Manager	<p>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.</p>
Auditor	<p>Users who are granted this role can view and modify the configuration settings for the security auditing subsystem.</p> <p>Note: The auditor role includes the monitor role. This allows the auditor to view but not change the rest of the security configuration.</p>

See Administrative roles in the WebSphere Application Server information center for more information.

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.

Tip: 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.

The failed event manager can be operated by any user granted either the administrator or the operator role.

Selectors can be configured by any user granted either the administrator or the configurator role

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 were not enabled.

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.

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. The security of these applications is vitally important.

Secure your applications by doing the following:

Procedure

1. Ensure that administrative security is enabled.
2. Ensure that application security is enabled.
 - a. On the administrative console, expand **Security** and click **Global security**.
 - b. Select **Enable application security** so that WebSphere Process Server will require authentication from users who try to access a secured application.
3. Develop your applications in WebSphere Integration Developer 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, so that the user need not repeatedly enter a user name and password.

Security in WebSphere Process Server can be divided 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 data accessed by an application is secured at origin, destination, and in transit:

- Integrity

Data sent over the network cannot 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, which is achieved through 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, which can, in turn, apply access control.

Authentication of users

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

Table 3. Authentication options for various clients

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

Table 3. Authentication options for various clients (continued)

Client	Authentication options	Notes
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. 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.

Default authentication aliases:

Several 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 predefined authentication aliases. Modify these aliases using the administrative console.

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

Table 4. Authentication aliases associated with business processes.

Alias	Description	Information
BPEAuthDataAliasJMS_node_server	Used to authenticate with the messaging engine.	Enter user name and password values in the applicable Business Process Choreographer properties in the response file.
BPEAuthDataAliasDbType_node_server	Used to authenticate with databases.	Configure the database using the provided scripts.

Table 5 describes the RunAs roles created for business processes.

Table 5. RunAs roles associated with business processes.

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

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

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

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

Common Event Infrastructure authentication aliases:

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

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

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

Alias	Description	Information
CommonEventInfrastructureJMSAuthAlias Note: 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.
EventAuthAliasDBType	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 alias:

The Service Component Architecture (SCA) has a predefined authentication alias. Modify the alias using the administrative console.

The alias in Table 7 is used to invoke the components regardless of the identity of the invoking user.

Table 7. Authentication alias 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.

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

Note: You can also access this page from various administrative console pages that require authentication alias information.

The authentication alias configuration page is displayed.

This page contains a list of authentication aliases, the associated component, the user ID associated with this alias, and optionally a description of the alias.

2. Select the authentication alias that you want to modify by clicking its name in the **Alias** column.

Note: In some cases, the **Alias** column might not provide a link, in which case you select the check box in the **Select** column corresponding to the alias that you want to edit, and click **Edit**.

3. Change the properties of the alias.

On the authentication alias configuration page 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 page, 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*

Access control

When a general user is authenticated to WebSphere 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 provide access control for components 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 Java EE role-based security. In contrast to code-based security, which secures the operation of components, role-based access control secures *resources*. For example, in the Business Calendars widget, you can specify the type of access that users have to individual timetables.

Security Roles widget

You use the Security Roles widget in Business Space to specify, for each timetable, the owner of the timetable as well as those who have writer and reader access to the timetable.

The following table shows the administrative roles and their default permission.

Roles	Default permission
BPMAdmin	Primary administrative user
BPMRoleManager	All authenticated users

EAR files and associated roles

The Business Process Choreographer and the Common Event Infrastructure are installed as part of WebSphere Process Server.

Table 8. EAR files and associated roles in WebSphere Process Server

Name of .ear file	Role	Default
BPEContainer_ <i>nodeName_serverName</i> .ear OR BPEContainer_ <i>clusterName</i>	APIUser	All Authenticated
	SystemAdministrator	None
	SystemMonitor	None
	JMSAPIUser	All Authenticated
	AdminJobUser	All Authenticated
	JAXWSAPIUser	Everyone
BPCEXplorer_ <i>nodeName_serverName</i> .ear OR BPCEXplorer_ <i>clusterName</i>	WebClientUser	All Authenticated
BSpaceEAR_ <i>nodeName_server</i> .ear	businessspaceusers	All Authenticated
BSpaceWebformsEnabler_ <i>nodeName_server</i> .ear	WebFormUsers	All Authenticated
BusinessRulesManager.ear	BusinessRuleUsers	All Authenticated
	NoOne	None
	AnyOne	Everyone
BusinessRules_ <i>nodeName_server</i> .ear	Administrator	All Authenticated
EventService.ear	eventAdministrator	All Authenticated
	eventConsumer	All Authenticated
	eventUpdater	All Authenticated
	eventCreator	All Authenticated
	catalogAdministrator	All Authenticated
	catalogReader	All Authenticated
mm.was_ <i>nodeName_server</i> .ear	All Authenticated	All Authenticated
	everyone	Everyone
REST Services Gateway.ear	RestServicesUser	All Authenticated
REST Services Gateway Dmgr .ear	RestServicesUser	All Authenticated

Table 8. EAR files and associated roles in WebSphere Process Server (continued)

Name of .ear file	Role	Default
TaskContainer_ <i>nodeNameserverName</i> .ear	APIUser	All Authenticated
	SystemAdministrator	None
OR TaskContainer_ <i>clusterName</i>	SystemMonitor	None
	EscalationUser	All Authenticated
	AdminJobUser	All Authenticated
	JAXWSAPIUser	Everyone
wpsFEMgr_7.0.0 Security	WBIOperator	Everyone

Business Process Choreographer Java EE roles

The following table lists Business Process Choreographer Java EE roles:

Table 9. Business Process Choreographer roles

Component	Roles	Value
BPEContainer	APIUser	All authenticated users
	SystemAdministrator	User names, group names, or both, entered during configuration
	SystemMonitor	All authenticated users
	JMSAPIUser	User name entered during configuration
	AdminJobUser	User name entered during configuration
	JAXWSAPIUser	Everyone
TaskContainer	APIUser	All authenticated users
	SystemAdministrator	SystemAdministrator
	SystemMonitor	SystemMonitor
	EscalationUser	EscalationUser
	AdminJobUser	AdminJobUser
	JAXWSAPIUser	Everyone

RunAs roles

In addition, applications make use of securityIdentity or RunAs roles as follows:

Table 10. The .ear files and associated RunAs roles

EAR file	J2EE Role
BPEContainer_ <i>nodeNameserverName</i> .ear	JMSAPIUser
	AdminJobUser
TaskContainer_ <i>nodeNameserverName</i> .ear	EscalationUser
	AdminJobUser

Access control in business process and human task applications:

Business Process Choreographer, which is installed as part of the WebSphere Process Server installation, uses roles to determine the capabilities of the user on a production system.

The Business Process Choreographer roles are shown in Table 11.

Table 11. Roles and default permissions

Roles	Default permission	Notes
System Administrator	User names, group names, or both, entered during configuration	Has access to all business processes and all operations.
System Monitor	All authenticated users	Has access to read operations.
JMSAPIUser	User name entered during configuration	All Business Process Choreographer JMS APIs are executed on behalf of this single user ID.
EscalationUser	User name entered during configuration	Used by the human task manager to process asynchronous API calls.
AdminJobUser	User name entered during configuration Note: The user supplied must be a member of the Business Process Choreographer System Administrator role.	Administrative jobs (for example, the cleanup service or business process instance migration) are executed on behalf of this single user ID.

Note: The WebClientUser role, which is associated with the Bpcexplorer.ear file, can access the Business Process Choreographer Explorer. The default permission for this role is All Authenticated.

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, refer to the WebSphere Application Server Network Deployment Information Center.

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.

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.

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 applications at the transport level for Web services* .

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 flows through multiple systems within the enterprise, the client must 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.

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 might 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) that 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 before 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 page 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.

Assigning users to roles

A secured application uses one or both of the security qualifiers `securityPermission` and `securityIdentity`. When these qualifiers are present, there are additional steps that 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 who is invoking the application.

Procedure

1. Follow the instructions for deploying an application into WebSphere Process Server. See [Deploying a module](#) for more details.
2. Associate the correct users with the roles.

Security qualifier	Action to take
Security Permission	<p>Assign a user or users to the role specified. There are four choices:</p> <ul style="list-style-type: none">• 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. <p>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.</p>
Security Identity	<p>Provide a valid user name and password for the identity to which the role is mapped.</p>

Commands to implement roles and user assignments (System Authorization Facility directions)

The System Authorization Facility (SAF) is a z/OS interface that programs can use to communicate with an external security manager, such as RACF. You can use RACF commands to implement roles and user assignments.

The following examples can be used to construct the RACF commands that are needed to implement the roles and user assignments:

```
RDEFINE EJBROLE (optionalSecurityDomain).WebClientUser UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).BPEAPIUser UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).BPESystemAdministrator UACC(NONE)
PERMIT (optionalSecurityDomain).BPESystemAdministrator CLASS(EJBROLE) ID(WSCFG1) ACCESS(READ)
RDEFINE EJBROLE (optionalSecurityDomain).BPESystemMonitor UACC(NONE)
PERMIT (optionalSecurityDomain).BPESystemMonitor CLASS(EJBROLE) ID(WSCFG1) ACCESS(READ)
RDEFINE EJBROLE (optionalSecurityDomain).JMSAPIUser UACC(READ) APPLDATA(RACFUserIdentity)
RDEFINE EJBROLE (optionalSecurityDomain).AdminJobUser UACC(READ) APPLDATA(RACFUserIdentity)
RDEFINE EJBROLE (optionalSecurityDomain).JAXWSAPIUser UACC(READ)
PERMIT (optionalSecurityDomain).JAXWSAPIUser CLASS(EJBROLE) ID(WSGUEST) ACCESS(READ)
RDEFINE EJBROLE (optionalSecurityDomain).businessspaceusers UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).WebFormUsers UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).BusinessRuleUsers UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).NoOne UACC(NONE)
RDEFINE EJBROLE (optionalSecurityDomain).AnyOne UACC(READ)
PERMIT (optionalSecurityDomain).AnyOne CLASS(EJBROLE) ID(WSGUEST) ACCESS(READ)
RDEFINE EJBROLE (optionalSecurityDomain).Administrator UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).RestServicesUser UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).TaskAPIUser UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).TaskSystemAdministrator UACC(NONE)
PERMIT (optionalSecurityDomain).TaskSystemAdministrator CLASS(EJBROLE) ID(WSCFG1) ACCESS(READ)
RDEFINE EJBROLE (optionalSecurityDomain).TaskSystemMonitor UACC(NONE)
PERMIT (optionalSecurityDomain).TaskSystemMonitor CLASS(EJBROLE) ID(WSCFG1) ACCESS(READ)
RDEFINE EJBROLE (optionalSecurityDomain).EscalationUser UACC(READ) APPLDATA(RACFUserIdentity)
RDEFINE EJBROLE (optionalSecurityDomain).Allauthenticated UACC(READ)
RDEFINE EJBROLE (optionalSecurityDomain).everyone UACC(READ)
PERMIT (optionalSecurityDomain).everyone CLASS(EJBROLE) ID(WSGUEST) ACCESS(READ)
RDEFINE EJBROLE (optionalSecurityDomain).WBIOperator UACC(READ)
PERMIT (optionalSecurityDomain).WBIOperator CLASS(EJBROLE) ID(WSGUEST) ACCESS(READ)
```

Any user who wants to make use of the applications protected by these roles must be granted Read access to the role. It is important to note that unsecured applications run under the identity of the WebSphere Application Server unauthenticated user ID, which by default is WSGUEST. This user ID is usually defined with the RESTRICTED option, so if an unsecured application uses application facilities protected by the Java EE roles listed above, then WSGUEST must be given read access to the relevant profiles that implement the equivalent of EVERYONE user mapping for the role.

Note: There is a subtlety in the user assignment to the roles when using SAF based authorization. To emulate EVERYONE access, the EJBROLE profile must be defined with a UACC of read and the WebSphere Application Server unauthenticated user ID (default WSGUEST) must be granted Read access. To emulate all authenticated access, the EJBROLE profile must be defined with a UACC of Read. For more information, see the WebSphere Application Server information center: System Authorization Facility considerations for the operating system and application levels.

Applications that use securityIdentity or RunAs roles also need extra configuration for SAF security products. In RACF, this is done by using the EJBROLE

APPLDATA parameter to assign a RACF user identity (RACFUserIdentity in the above examples) to the role. For more information, see the WebSphere Application Server information center: System Authorization Facility (SAF) delegation.

Security for the Business Calendars widget

The Security Roles widget provides you with the ability to secure access to individual timetables in the Business Calendars widget. You use the Security Roles widget to assign roles to the members of an organization. It is these roles that determine the level of access to the timetables.

The Security Roles widget, which you use to administer role-based access control for the Business Calendars widget, is located in Business Space powered by WebSphere.

This role-based access is based on XACML (eXtensible Access Control Markup Language), an open standard.

What are the advantages of using the Security Roles widget role-based access control in the Business Calendars widget?

- You can control access to a specific instance of a timetable.
For example, you can specify that a user has access only to the user's own timetable and that the user does not have the ability to look at or change anyone else's timetable.
- Controlling access is done at the role level, instead of the individual user level.
You map members to roles. It is the role that defines the permission members have to the specific instance of the resource.

Roles associated with a timetable

When a timetable is installed, three roles are created for that timetable—Owner, Writer, and Reader. These roles are known as component-specific roles.

How would these roles be used? Consider the case of a holiday timetable used in an organization. You want all employees to have access to the timetable, but you want to limit the number of employees who can update the timetable.

When the Holiday timetable is installed, the following roles are created:

- **HolidayOwner**
Members assigned to this role can read the Holiday timetable and can also write to it. For example, if the company decided to add an extra holiday, a member with the HolidayOwner role would be able to make the change.
Members of this role can also assign members to the HolidayWriter and HolidayReader role. For example, the HolidayOwner might decide to add a senior manager to the HolidayWriter role.
- **HolidayWriter**
Members assigned to this role can read the Holiday timetable and can also write to it. As in the case of the HolidayOwner, members of the HolidayWriter role could add the extra holiday.
- **HolidayReader**
Members assigned to this role can read the Holiday timetable but cannot write to it.

You might assign the HolidayOwner role to the Human Resources manager, the HolidayWriter role to the Human Resources Specialists group, and the

HolidayReader role to the employee group, as shown in the following figure:

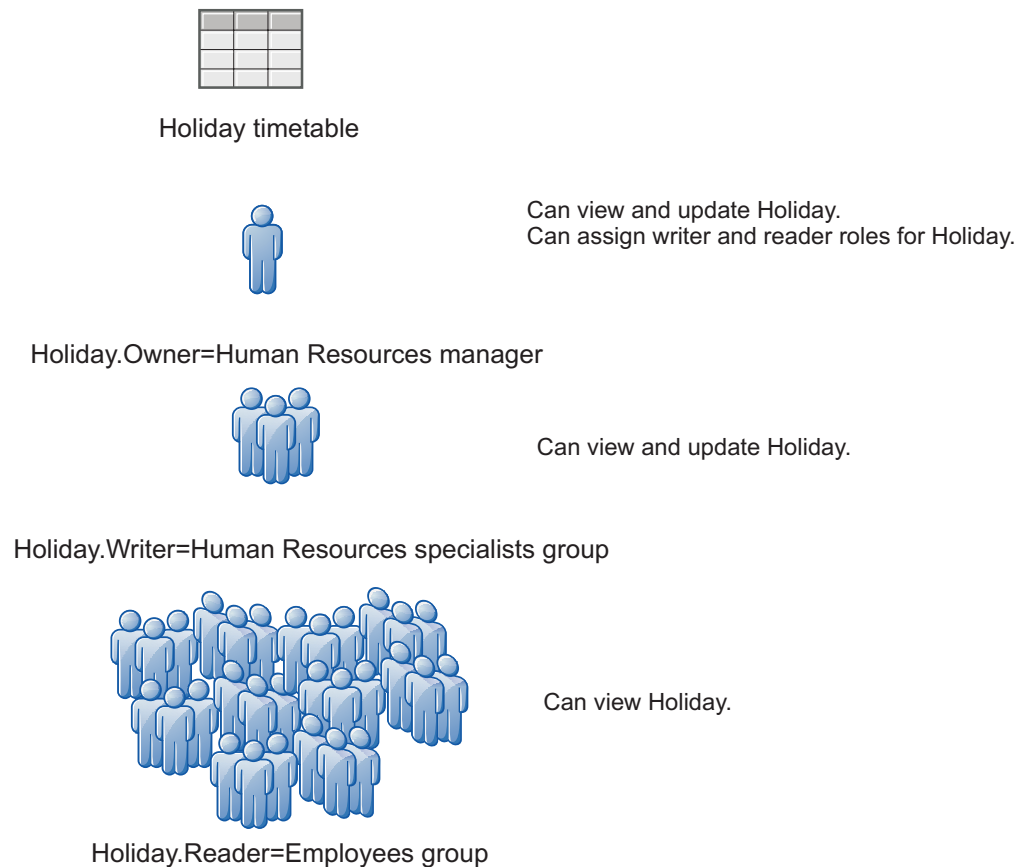


Figure 1. Example of roles assigned to a timetable

When you deploy a timetable, the three roles—Owner, Writer, and Reader—are created. Permission for all roles is set initially to **All Authenticated**. Make sure that you change this designation to assign the members of the organization to the correct roles.

Note: You can change the membership of a role (for example, you can remove a member from the reader role), but you cannot change the name of a role, add or delete a role, or change the permissions associated with a role. The permissions are set as follows:

- Members of the Owner role can read and write to the timetable and can assign other members to the Writer and Reader roles.
- Members of the Writer role can read and write to the timetable.
- Members of the Reader role can read the timetable.

In the Security Roles widget, these timetable-related roles are also known as *module roles*.

System roles for the Security Roles widget

The BPMAAdmin and BPMRoleManager roles are automatically created when you enable security after installing WebSphere Process Server (or upgrading to WebSphere Process Server 7.0).

- BPMAAdmin

BPMAdmin has the authority to add members to or remove members from the BPMRoleManager role. For example, if the person performing the BPMRoleManager role leaves the organization, only BPMAdmin can assign another member to that role.

BPMAdmin is initially assigned to one member—the primary administrative user. Change this assignment to another member as soon as you restart the server after installation or upgrade.

- **BPMRoleManager**

BPMRoleManager has the authority to add members to or remove members from the three timetable-related roles—Owner, Writer, and Reader. For example, if a Holiday timetable is created, the BPMRoleManager assigns members to the HolidayOwner, HolidayWriter, and HolidayReader roles.

BPMRoleManager is initially assigned to one member—the primary administrative user. Change this assignment to another member as soon as you restart the server after installation or upgrade.

Administering roles in the Security Roles widget

Using the Security Roles widget, you can assign a user or group to the system roles. You can also assign a user or group to the component roles associated with timetables.

Assigning component roles

Each timetable in the Business Calendars widget has three component roles—Owner, Writer, and Reader—associated with it. You use the Security Roles widget to assign users or groups to these roles.

Before you begin

Make sure the Security Roles widget is displayed.

About this task

The BPMRoleManager can assign users or groups to component roles.

The Owner of a timetable can also assign users or groups to the Owner, Writer, or Reader role for that timetable.

Procedure

1. To assign individual members to a module role, complete the following steps:
 - a. From the **Module** list, select a timetable.
 - b. For a role (for example, the Writer role for the timetable), click the name of the role.
 - c. On the right side of the page, click **Add**.
 - d. Type a name (or part of a name) in the **Users or groups to search for** field.
 - e. To restrict the number of users or groups returned based on the search criteria, change the value in the **Maximum result** field. Set this value to 0 to return the entire result set.
 - f. Click **Search**.
 - g. From the list that is displayed, select one or more users or groups and click **OK**.
 - h. When you have assigned all members, click **Save**.
2. To assign all members to a module role, complete the following steps:

- a. From the **Module** list, select a timetable.
- b. For a role (for example, the Reader role for the timetable), click the name of the role.
- c. Select **All authenticated**.
- d. Click **Save**.

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

An adapter is the mechanism by which an application communicates with an Enterprise Information System (EIS). 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 an EIS and Java EE 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. JCA 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 provides information about how to secure this link.

Setting up security for Business Space

If you are using Business Space powered by WebSphere with your environment, you must consider security options for how your team will work with artifacts in Business Space. If you want to turn on security for Business Space, set up application security and designate a user repository. To define Business Space administrators, assign a superuser role.

About this task

For best results, enable security before you configure Business Space. On the administrative console Global security administration page, you enable both administrative security and application security. You also designate a user account repository.

Considerations for using a user account registry with Business Space:

- Based on the type of LDAP configuration that you are using, your settings can impact your ability to access Business Space correctly. Make sure that the user filters, the group filters, and mapping settings are configured properly. For more information, see *Configuring Lightweight Directory Access Protocol search filters* in the WebSphere Application Server documentation.
- Based on the type of federated repository configuration that you are using, your settings can affect your ability to access Business Space correctly. Make sure that the realms are configured properly. For more information, see *Managing the realm in a federated repository configuration* in the WebSphere Application Server documentation.
- The LDAP security is set up by default to use the login property `uid` (user ID) for searching in Business Space. If your LDAP security is changed to use another unique LDAP field, such as `mail` (e-mail address) for the login property, then you must modify the `userIdKey` property in the `ConfigServices.properties` file in order for searching to work in Business Space. The `ConfigServices.properties` file is located at `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties` for a stand-alone server or `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties` for a cluster. Change the `userIdKey` attribute from `uid` to match the login property for your LDAP security, for example, `mail`. Then run the `updatePropertyConfig` command using the `wsadmin` scripting client, designating the following parameters: **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster, **-propertyFileName** with the value of the path for the `ConfigServices.properties` file, and **-prefix** with the value `Mashups_`.
- If you are using a Microsoft® SQL Server database and the **Standalone LDAP** registry, make sure that the user distinguished name (user DN) does not exceed 131 characters. If any of the user DN entries exceed 131 characters, you must designate the **Federated repositories** option for the user account repository. When switching between federated repositories and other registries, all the existing spaces, pages are no longer accessible in Business Space and must be created again.

- If you are using **Federated repositories**, you have additional capabilities in your widgets and framework, such as enhanced search capabilities. When searching for users to share spaces and pages, the search scope includes e-mail, a full user name, and user ID.

If you are using IBM® Tivoli Access Manager WebSEAL and want to use it with your Business Space environment, you must complete additional configuration steps. Configure Tivoli Access Manager security with an external Java Authorization Contract for Containers (JACC) provider, configure WebSEAL with Tivoli Access Manager, configure WebSEAL with your product application server, and configure host junctions for your environment.

To set up which users in the Business Space environment will be administrators, you run a script to assign the Business Space superuser role.

Setting application security for Business Space

To turn on security for Business Space you must enable both application security and administrative security.

Before you begin

Before you complete this task, you must have completed the following tasks:

- Checked that your user ID is registered in the user registry for your product.

If you expect to use a secured environment, make sure to enable security before you configure Business Space. If you want to enable or remove security after you have configured Business Space, you must modify both the `MashupAdminForOOBSpace` property and the `noSecurityAdminInternalUserOnly` property in the `ConfigServices.properties` file to set the correct user ID as the valid administrator ID. The `ConfigServices.properties` file is located at `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties` for a stand-alone server or `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties` for a cluster. Copy the modified file into an empty folder on your system. Then run the `updatePropertyConfig` command using the `wsadmin` scripting client, designating the following parameters:

- **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster
- **-propertyFileName** with the value of the path for the `ConfigServices.properties` file
- **-prefix** with the value `Mashups_`

About this task

Business Space is preconfigured to ensure authentication and authorization of access. Users are prompted to authenticate when accessing Business Space URLs. Unauthenticated users are redirected to a login page. Business Space can be accessed by either HTTP or HTTPS, except for the login page, which always redirects to HTTPS. Therefore if using a Web server such as the IBM HTTP Server, you must configure it to support HTTPS.

Authorization to spaces and page content in Business Space is handled internally to Business Space as part of managing spaces.

To enable authenticated access to Business Space, you must have a user registry configured and application security enabled.

Procedure

1. For complete instructions on security, see the security documentation for your product.
2. For the Business Space application, on the Global security administrative console page, select both **Enable administrative security** and **Enable application security**.
3. On the same administrative console page, under **User account repository**, designate either **Federated repositories**, **Local Operating System**, **Standalone LDAP registry**, or **Standalone custom registry**. Review the considerations for selecting a user registry in Setting up security for Business Space.
4. If Business Space is remote from where your product is running, and if the node where Business Space is running and the node where your product is running are not in the same cell, you must complete manual steps to make sure that single-sign-on (SSO) is enabled. For example, if you are using more than one product (WebSphere Business Compass, WebSphere Business Monitor, WebSphere Enterprise Service Bus, or WebSphere Process Server), the servers are on different nodes, and you want them all to be able to work with the Business Space server, you must manually configure SSO. To enable SSO, complete the following steps:
 - a. On the administrative console for each server, open the Global security page by clicking **Security** → **Global security**. Expand **Web and SIP security** and click **single sign-on (SSO)** to make sure that the **Enabled** check box is selected.
 - b. Make sure that all the nodes use the same **User account repository** information (see step 3).
 - c. On the administrative console for the first node, open the Global security page. Under Authentication, click **LTPA**.
 - d. Under **Cross-cell single sign on**, type a password for the key file and a fully qualified key file name, which is a location and file name where you want to export the key file. The fully qualified key file name is the absolute path on the system where your server is running.
 - e. Click **Export keys**. The key file is saved on the system where the server is running.
 - f. If the two nodes are not on the same system, copy the key file physically to the other systems.
 - g. Import the key file on every other node using the same key file: Log on to the administrative console for the other nodes, and go to the Global security > LTPA page. Under **Cross-cell single sign on**, type the password for the key file and the fully qualified key file name (use the same password for the exported key file that you copied over), and click **Import keys**.
 - h. Restart the server after importing keys on each system.
5. If you are using HTTPS in the endpoints file, the endpoint location is on a different node than Business Space, and the Secure Sockets Layer (SSL) certificate is a self-signed SSL certificate, you must import it.
 - a. Log on to the administrative console for the server that contains Business Space and import the SSL certificate that is used by the remote node where product is running.
 - 1) Under Security, click **SSL certificate and key management**.

- 2) On the SSL certificate and key management page, under Related items, click **Key stores and certificates**.
 - 3) On the Key stores and certificates page, click **NodeDefaultTrustStore** to modify that truststore type.
 - 4) On the NodeDefaultTrustStore page, under Additional Properties, click **Signer certificates**.
 - 5) On the Signer certificates page for the **NodeDefaultTrustStore**, click the **Retrieve from port** button.
 - 6) On Retrieve from port page, under General Properties, type the host, port, and alias for where your product is running. Click **Retrieve signer information** button and then click **OK**.
 - 7) Restart both servers.
- b. Log on to the administrative console for the product node and import the SSL certificate that is used by the node where Business Space is running.
- 1) Repeat steps a. i.-v.
 - 2) On the Retrieve from port page, under General Properties, type the host and port for where Business Space is running. Click the **Retrieve signer information** button and then click **OK**.
 - 3) Restart both servers.

For more information about SSO and SSL, see the WebSphere Application Server information center.

What to do next

- After the administrative security and application security are turned on, you receive a prompt for a user ID and password when you log on to Business Space. You must use a valid user ID and password from the selected user registry in order to log on. After you turn on administrative security, whenever you return to the administrative console, you must log on with the user ID that has administrative authority.
- If you want to restrict logging in to Business Space to a subset of users and groups, you can change the mapping of the Business Space J2EE role. You must update the user/group mapping for two enterprise applications: **BSpaceEAR_node_server** and **mm.was_node_server**. Click **Applications** → **Application Types** → **WebSphere enterprise applications** and select the two applications. In the right panel, under Detail Properties, select **Security role to user/group mapping**. Remap the **businessspaceusers** and **Allauthenticated** roles from the two applications by first removing the special subject. Click **Map Special Subjects** and select **None**. Then click **Map Users** or **Map Groups** and assign each role to your selected users or groups. Note that changing the J2EE role mapping does not affect the user/group search function in Business Space.
- To set authorization to pages and spaces in Business Space, you can manage authorization when you create Business Space pages and spaces.
- **Monitor** **Process Server / ESB** To set up security for the data in the widgets based on users and groups, you must modify the mapping of users to the REST services gateway application. Select the REST services gateway application, and in the right panel, under Detail Properties, select **Security role to user/group mapping**. For the RestServicesUser role, you can add users and groups to it to control access to the data in all the REST services widgets.
- **Process Server / ESB** If you want to restrict access to data in the widgets based on user group roles, consider changing the users assigned to the administrative group roles. You can view the Roles list to see who is assigned to

these roles by opening the administrative console, clicking **Security** → **Secure administration, applications, and infrastructure** → **Administrative Group Roles**, and selecting a group.

You might want to consider changing the users assigned to administrative group roles for widgets such as Business Rules and Business Variables.

For example, for the System Health widget, the following administrative roles all have monitoring permissions, all allow access to the administrative console, and therefore allow users assigned to those roles to access data in the System Health widget:

- **Monitor**
- **Configurator**
- **Operator**
- **Administrator**
- **Adminsecuritymanager**
- **Deployer**
- **iscadmins**

Users who are mapped to those administrative group roles have access to the data in the System Health widget. Users who are not mapped to those roles cannot access the data in the System Health widget.

- Finally, some widgets have an additional layer of role-based access for their artifacts created by business users. For WebSphere Process Server administration widgets, the Security Roles widget allows you to assign users and groups to system roles or module roles that determine the level of access that members have for timetables in the Business Calendars widget. For WebSphere Business Compass, the Review Access Control widget manages permissions for users who can review and comment on reviews. For more information, see the online help for your widgets.

Note:

If you find the following errors in the SystemOut.log file, you might have extra attributes in your user registry that cannot be processed:

```
00000046 SystemErr R Caused by: com.ibm.websphere.wim.exception.WIMSystemException: CWWIM1013E
    The value of the property secretary is not valid for entity uid=xxx,c=us,ou=yyy,o=ibm.com.
00000046 SystemErr R at com.ibm.ws.wim.adapter.ldap.LdapAdapter.setPropertyValue(LdapAdapter.java:3338)
```

Set the following attributes in the ConfigServices.properties file to bypass those attributes:

```
com.ibm.mashups.user.userProfile = LIMITED
com.ibm.mashups.user.groupProfile = LIMITED
```

The ConfigServices.properties file is located at *profile_root*\BusinessSpace*node_name*\server_name\mm.runtime.prof\config\ConfigService.properties for a stand-alone server or *deployment_manager_profile_root*\BusinessSpace*cluster_name*\mm.runtime.prof\config\ConfigService.properties for a cluster. After modifying the ConfigServices.properties file, run the updatePropertyConfig command using the wsadmin scripting client, designating the following parameters: **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster, **-propertyFileName** with the value of the path for the ConfigServices.properties file, and **-prefix** with the value Mashups_.

Note:

If you have Java 2 security enabled in a cluster, consider tightening the entry in the server policy applied to the Business Space help location.

The Business Space help location policy is:

```
grant codeBase      "file:${was.install.root}/profiles/profile_name/temp/
node_name/-" {

    permission java.security.AllPermission;

};
```

Tighten the policy by changing it to:

```
grant codeBase      "file:${was.install.root}/profiles/profile_name/temp/
node_name/server_name/BusinessSpaceHelpEAR_node_name_server_name/
BusinessSpaceHelp.war/-" {

    permission java.security.AllPermission;

};
```

Configuring Tivoli Access Manager WebSEAL to work with Business Space

If you have Tivoli Access Manager WebSEAL and you want to use it with Business Space, you must complete several additional configuration steps.

About this task

If you want to use Tivoli Access Manager WebSEAL with Business Space, you must configure Tivoli Access Manager security with an external Java Authorization Contract for Containers (JACC) provider, configure WebSEAL with Tivoli Access Manager, configure WebSEAL with your product application server, and configure host junctions for your environment.

Procedure

1. Configure Tivoli Access Manager with JACC.
 - a. Complete one of the following steps, depending on whether you want to use the administrative console or the wsadmin commands.
 - If you want to use the administrative console to configure Tivoli Access Manager with JACC, complete the following steps:
 - 1) Enable Global Security.
 - a) Select **Security** → **Global Security**.
 - b) Enable **Administrative security**, **Application security**, and **Java 2 security** with the LDAP server with which Tivoli Access Manager is configured.
 - c) Select **Global Security** → **LDAP**, enter the following information, and then click **OK**.

Name	Description
Server user Id	Enter the same user ID that you entered for the administrator DN on Tivoli Access Manager settings. Example: user1

Name	Description
Server user password	puser1
Host	LDAP configured with Tivoli Access Manager
Port	Example: 389
Base DN	Example: o=ibm, c=us
Bind DN	Example: cn=SecurityMaster,secAuthority=Default
Bind pwd	password for SecurityMaster user

- d) Save the configuration, and restart the server.
- 2) Enable external authorization with Tivoli Access Manager and JACC.
 - a) Select **Security** → **Global Security** → **External authorization providers**.
 - b) In the **Authorization provider** list, select **External JACC provider**, and then click **Configure**. The default properties for Tivoli Access Manager are correct. For default values, do not change.
 - c) Under **Additional Properties**, select **Tivoli Access Manager properties**. Select **Enable embedded Tivoli Access Manager**, enter the following information, and then click **OK**.

Name	Value
Client listening port set	The default setting is 8900 - 8999. Change it only if you want to use different ports.
Policy server (name:port)	Specify your <i>policyserver:port</i> . Example: windomain3.rtp.raleigh.ibm.com:7135
Authorization servers and priority (name:port:priority)	Specify your <i>authorizationserver:port:priority</i> . Example: windomain3.rtp.raleigh.ibm.com:7136:1
Administrator user name	Leave the user name as sec_master (default) , unless you use a different admin name on the Tivoli Access Manager server.
Administrator user password	domino123
User registry distinguished name suffix	Type the name that you want to use for your application server. Example: o=ibm,c=us
Security domain	Leave the Security domain set to Default . Change this setting if you are not using the default domain on the Tivoli Access Manager server. Change this setting if you have multiple domains created on the Tivoli Access Manager server and you want to connect or use a domain other than Default .
Administrator user distinguished name	Type the fully qualified name of the user. Example: cn=user1,o=ibm,c=us Note: This user is the same as the Server user ID configured in the LDAP user registry panel.

The server contacts the Tivoli Access Manager server and creates several properties files under the application server. This process might take a few minutes. If an error occurs, look in system Out and correct the problem.

- If you want to use the wsadmin utility to configure Tivoli Access Manager with JACC, complete the following steps. Perform the following procedure once on the deployment manager server. The configuration parameters are forwarded to managed servers, including node agents, when a synchronization is performed. The managed servers require their own restart for the configuration changes to take effect.

- 1) Verify that all the managed servers, including node agents, are started.
- 2) Start the server.
- 3) Start the command-line utility by running the wsadmin command from the *install_root/bin* directory.
- 4) At the wsadmin prompt, run the configureTAM command, including the appropriate information from the following table:

Jacl example:

```
$AdminTask configureTAM -interactive
```

Jython example:

```
AdminTask.configureTAM('-interactive')
```

Then type the following information:

Name	Value
node name for your product server	Specify a single node or enter an asterisk (*) to choose all nodes.
Tivoli Access Manager Policy Server	Type the name of the Tivoli Access Manager policy server and the connection port. Use the format, <i>policy_server:port</i> . The policy server communication port is set at the time of Tivoli Access Manager configuration. The default port is 7135.
Tivoli Access Manager Authorization Server	Type the name of the Tivoli Access Manager authorization server. Use the format <i>auth_server:port:priority</i> . The authorization server communication port is set at the time of Tivoli Access Manager configuration. The default port is 7136. You can specify more than one authorization server by separating the entries with commas. Having more than one authorization server configured is useful for failover and performance. The priority value is the order of authorization server use. For example: <i>auth_server1:7136:1,auth_server2:7137:2</i> . A priority of 1 is still required when configuring against a single authorization server.
administrator distinguished name for your product server	Type the full distinguished name of the security administrator ID for your product server. For example: <i>cn=wasadmin,o=organization,c=country</i> . For more information, see the related link.

Name	Value
Tivoli Access Manager user registry distinguished name suffix	For example: o=organization, c=country
Tivoli Access Manager administrator user name	Type the Tivoli Access Manager administration user ID, as created at the time of Tivoli Access Manager configuration. This ID is typically sec_master.
Tivoli Access Manager administrator user password	Type the password for the Tivoli Access Manager administrator.
Tivoli Access Manager security domain	Type the name of the Tivoli Access Manager security domain that is used to store users and groups. If a security domain is not already established at the time of Tivoli Access Manager configuration, click Return to accept the default.
Embedded Tivoli Access Manager listening port set	The product server listens on a TCP/IP port for authorization database updates from the policy server. Because more than one process can run on a particular node and machine, a list of ports is required for the processes. Specify the ports that are used as listening ports by Tivoli Access Manager clients, separated by a comma. If you specify a range of ports, separate the lower and higher values by a colon. For example, 7999, 9990:9999.
Defer	Set to yes, this option defers the configuration of the management server until the next restart. Set to no, configuration of the management server occurs immediately. Managed servers are configured on their next restart.

- 5) After you enter all the required information, select **F** to save the configuration properties or **C** to cancel from the configuration process and discard the entered information.

Example with SVTM TAM60 server:

```
wsadmin>$AdminTask configureTAM -interactive
Configure embedded Tivoli Access Manager
```

This command configures embedded Tivoli Access Manager on the WebSphere Application Server node or nodes specified.

```
WebSphere Application Server Node Name (nodeName): *
*Tivoli Access Manager Policy Server (policySvr):
  windomain3.rtp.raleigh.ibm.com:7135
*Tivoli Access Manager Authorization Servers (authSvrs):
  windomain3.rtp.raleigh.ibm.com:7136:1
*WebSphere Application Server administrator's distinguished name (wasAdminDN):
  cn=was61admin,o=ibm,c=us
*Tivoli Access Manager user registry distinguished name suffix (dnSuffix):
  o=ibm,c=us
Tivoli Access Manager administrator's user name (adminUid):
  [sec_master]
*Tivoli Access Manager administrator's user password (adminPasswd):
  domino123
Tivoli Access Manager security domain (secDomain): [Default]
Embedded Tivoli Access Manager listening port set (portSet): [9900:9999]
Defer (defer): [no]
```

Configure embedded Tivoli Access Manager

F (Finish)
C (Cancel)

Select [F, C]: [F] F

```
WASX7278I: Generated command line: $AdminTask configureTAM {-policySvr
windomain3.rtp.raleigh.ibm.com:7135 -authSvrs
windomain3.rtp.raleigh.ibm.com:7136:1 -wasAdminDN cn=wa
Embedded Tivoli Access Manager configuration action parameters saved successfully.
Restart all WebSphere Application Server instances running on the target node or
nodes to
wsadmin>
```

- 6) In the administrative console, select **Security** → **Global Security** → **External authorization providers**. Then select **External authorization using a JACC provider**, and click **OK**.
 - 7) Go to the main security screen and click **OK**. Save and synchronize your changes.
 - 8) Restart all processes in your cell.
- b. If you installed applications before you enabled Tivoli Access Manager (for example, you enabled LDAP security and installed some secured applications and mapped users and groups to security roles), propagate the security roles mapping information from the deployment descriptors to the Tivoli Access Manager policy server. Perform one of the following steps, depending on whether you want to use the administrative console, or the wsadmin commands.
- If you want to use the propagatePolicyToJACCProvider wsadmin command, see Propagating security policy of installed applications to a JACC provider using wsadmin scripting.
 - If you want to use the administrative console, see Propagating security policies and roles for previously deployed applications.
2. Configure WebSEAL with Tivoli Access Manager.
- a. Ensure that WebSEAL is installed and configured properly.
 - b. Create the junction between WebSEAL and your product application server using the **-c iv_creds** option for TAI++ and **-c iv_user** for TAI. Enter either of the following commands as one line, using the variables that are appropriate for your environment:
For TAI++
server task webseald-server create -t tcp -b supply -c iv_creds
-h *host_name* -p *websphere_app_port_number* *junction_name*
 - c. To create a trusted user account in Tivoli Access Manager, which can be used for configuring TAI, issue the following commands:
pdadmin -a sec_master -p domino123
pdadmin sec_master> user create -gsouser -no-password-policy taiuser
"cn=taiuser,ou=websphere,o=ibm,c=us" taiuser taiuser ptaiuser
pdadmin sec_master> user modify taiuser password-valid yes
pdadmin sec_master> user modify taiuser account-valid yes
 - d. In the WebSEAL configuration file *webseal_install_directory/etc/webseald-default.conf*, set the following parameter:
basicauth-dummy-passwd=*webseal_userid_passwd*
For example, if you set the taiuser/ptaiuser in Tivoli Access Manager, set the following parameter:basicauth-dummy-passwd = ptaiuser

If you are using a form-based authentication, set the following parameters:

forms-auth=both

ba-auth=none

3. Configure WebSEAL with your product application server by enabling the TAI++ interceptor on the server.
 - a. In the administrative console, select **Global security** → **Authentication mechanisms and expiration**.
 - b. Expand **Web and SIP security**, and then select **Trust Association**. Select the check box and click **Apply**.
 - c. Select **Interceptors** → **TAMTrustAssociationInterceptorPlus** → **custom properties**, and add the following properties:

Name	Value
com.ibm.websphere.security.webseal.configURL	\${WAS_INSTALL_ROOT}/java/jre/PdPerm.properties
com.ibm.websphere.security.webseal.id	iv-creds
com.ibm.websphere.security.webseal.loginId	taiuser (if the user taiuser/ptaiuser was created in the Tivoli Access Manager)

- d. Restart the cell.
 - e. To access the client, go to `https://webseal_server_name:webseal_port/junction_name/web_uri_for_client`.
4. Configure the host junctions for your environment, so that the Business Space widgets appear. Complete one of the following steps, depending on whether you are using virtual host junctions or transparent host junctions.
 - If you are using virtual host junctions, create a virtual host junction. A virtual host junction eliminates the need to create separate junctions.
 - a. Make sure that a virtual host has been configured. Virtual host junctions match a host and port number and forward addresses to the target host. No URL filtering occurs, and all requests that match are forwarded to the target host.
 - b. Make sure that the following applications are available to the same virtual host. You may have some or all of the applications, based on which products you are using with Business Space.
 - BPMAAdministrationWidgets_nodename_servername (for WebSphere Enterprise Service Bus and WebSphere Process Server)
 - BusinessSpaceHelpEAR_nodename_servername (for all products)
 - BSpaceEAR_nodename_servername (for all products)
 - BSpaceWebformsEnabler_nodename_servername (for all products)
 - HumanTaskManagementWidgets_nodename_servername (for WebSphere Process Server and WebSphere Business Monitor)
 - REST Services Gateway (for all products)
 - REST Services Gateway Dmgr (for WebSphere Enterprise Service Bus and WebSphere Process Server)
 - mm.was_nodename_servername (for all products)
 - WBMDashboardWeb_nodename_servername (for WebSphere Business Monitor)
 - wesbWidgets_nodename_servername (for WebSphere Enterprise Service Bus)
 - widgets_busleader_nodename_servername (for WebSphere Business Compass)

- `widgets_pubserver_nodename_servername` (for WebSphere Business Compass)
- `widgets_fabric_nodename_servername` (for WebSphere Business Services Fabric)

Note: This list of applications covers only the applications required by Business Space. You might need to add other applications to the list for non-Business Space scenarios using Tivoli Access Manager WebSEAL.

- c. Run the following command using `pdadmin`: `server task webseal server virtualhost create -t transport -h target_host [-p port] [-v virtual_host_name] virtual_host_label`

Use the following information:

- `webseal server` is the name of the WebSEAL server where you will create the virtual host entry.
- `transport` is the type of transport. Valid entries are `tcp`, `ssl`, `tcpproxy`, and `sslproxy`.
- `target_host` is the host of the required application.
- `virtual_host_name` is used to match HTTP requests to a virtual host junction. If no value is entered, it is made up of the target host and port by default. For example, if you set the `virtual_host_name` to `myvirthost.ibm.com:80`, WebSEAL matches the URLs containing `myvirthost.ibm.com:80` and routes it to the host provided in the `pdadmin` command.
- `virtual_host_label` is the label used to identify the entry in WebSEAL. It must be unique.

For Business Space to run as expected, both `ssl` and `tcp` entries must be created for the type of transport. When you need both Secure Sockets Layer (SSL) and Transmission Control Protocol (TCP) to be supported in the same virtual host junction, you must use the `-g vhost_label` option, where `vhost_label` is the original virtual host label to share configuration. This option finds a previously created virtual host junction (one created earlier, where the `virtual_host_label` matches the label provided in the `-g` option), and will share that configuration. The second entry still needs its own `virtual_host_label`, but it can share the target host, port, and other values. If you do not provide this `-g` option, a second virtual host cannot be created because WebSEAL will see the target host and port as being identical to a previously create junction (which is not allowed).

- If you are using transparent host junctions, create a series of transparent path junctions for the widgets for each product.
 - a. Run the following command using `pdadmin`: `server task webseal server create -t transport type (ssl) or (tcp) -x -h hostname path`
For example, type: `server task webseald-default create -t tcp -x -h monServer.ibm.com /BusinessSpace`.
 - b. Create the following context roots for your product: Mapping Business Space URLs for a reverse proxy server.
5. Complete additional configuration steps to resolve issues with browser cookies and virtual hosts.
 - a. To resolve renaming of the Business Space cookie, add the following content to the WebSEAL configuration file:


```
[preserve-cookie-names]
name = com.ibm.bspace.UserName
```

```
name = com.ibm.wbimonitor.UserName
```

- b. Optional: If you are using non-default virtual hosts with a context root, you might encounter issues with Business Space pages. You might need to stop the junction from rewriting the JavaScript™ on the Business Space pages by adding the -j junction to the context root. Run the following command:
server task default-webseald create -f -h *hostname* -p *portnumber* -t tcp -b supply -c iv-user,iv-creds,iv-groups -x -s -j -J trailer/root *context*

Assigning the Business Space superuser role

In Business Space, you can assign users to be superusers (or Business Space administrators). A superuser can view, edit, and delete all spaces and pages, can manage and create templates, and can change ownership of a space by changing the owner ID.

Before you begin

If administrative security is enabled when you configure Business Space, consider the following information about groups and superusers:

- Users belonging to the special user group, **administrators**, have a superuser role by default. As a result, the superuser role assignment is handled by user group membership.
- In a single-server environment, the Business Space server creates the **administrators** user group in the default user registry. The administrator ID provided during configuration is automatically added as member of this group.
- In a network deployment environment, the **administrators** user group is not created automatically. Use the `createSuperUser.py` script to create the user group and add members to that group in the default user registry.
- If another user registry (for example, LDAP) is used instead of the default user registry, or if the default user registry is used but you do not want to use the **administrators** user group, you must identify the user group that you are using for the Business Space superusers. Make sure that the value you provide can be understood by the user registry. For example, for LDAP, you might provide a name like `cn=administrators,dc=company,dc=com`. For more information about identifying this user group, see the instructions for changing the administrators group in the What to do next section.
- For Business Space in WebSphere Portal, the default group **wpsadmins** is also used for the superuser role. Members of this group are granted the superuser role for Business Space.

Note: Security must be enabled if you want to use Business Space in WebSphere Portal.

If administrative security is not enabled when you configure Business Space, only the special user ID **BPMAdministrator** has the Business Space superuser role.

If you have a network deployment environment, you must run the `createSuperUser.py` script to assign the superuser role: to create the user group and add members. Before you run the script, complete the following steps:

- Make sure the default **administrators** group name is not changed.
- Use the default repository for the user registry.
- Start the server or the deployment manager for your Business Space environment for the profile where is Business Space installed.

Procedure

1. Locate the script `install_root\BusinessSpace\scripts\createSuperUser.py` for assigning the superuser role to a user.
2. Open a command prompt, and change directories to the following directory: `profile_root\bin`, where `profile_root` represents the directory for the profile where Business Space is installed.
3. Type the following command: `wsadmin -lang jython -f install_root\BusinessSpace\scripts\createSuperUser.py user_short_name password` where `user_short_name` is the unique identifier for a user in Virtual Member Manager (VMM), and `password` is the VMM password for that user. If that user exists in VMM, the user is added to the administrator group.

Note: When the path contains a space, for example, if `install_root` is `My install dir`, you must enclose the path names in quotation marks. For example, type the following command: `wsadmin -lang jython -f "\My install dir\BusinessSpace\scripts\createSuperUser.py" user_short_name_in_VMM`.

What to do next

To open Business Space, use the following URL: `http://host:port/BusinessSpace`, where `host` is the name of the host where your server is running and `port` is the port number for your server.

You can change the default special user group named **administrators**. Perform the following steps to check the current group name or change it to other name.

Inspect the value for the metric `com.ibm.mashups.adminGroupName` in the configuration file:

- `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties` on a stand-alone server, or
- `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties` on a cluster.

If you want to change an administrative group, perform the following steps on a stand-alone server:

1. Modify the metric `com.ibm.mashups.adminGroupName` in the configuration file `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties`.
2. Run the command `updatePropertyConfig` in the `wsadmin` environment of the profile: `$AdminTask updatePropertyConfig {-serverName server_name -nodeName node_name -propertyFileName "profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties" -prefix "Mashups_"} and run $AdminConfig save.`
3. Restart the server.

If you want to change an administrative group, perform the following steps on a cluster:

1. Modify the metric `com.ibm.mashups.adminGroupName` in the configuration file `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties`.
2. Run the command `updatePropertyConfig` in the `wsadmin` environment of the deployment environment profile: `$AdminTask updatePropertyConfig {-clusterName cluster_name -propertyFileName`

```
"deployment_manager_profile_root\BusinessSpace\cluster_name\  
mm.runtime.prof\config\ConfigService.properties" -prefix "Mashups_" and  
run $AdminConfig save.
```

3. Restart the deployment manager.

If you want to change the superuser when security is not enabled, perform the following steps on a stand-alone server:

1. Modify the metric `noSecurityAdminInternalUserOnly` in the configuration file `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties`.
2. Run the command `updatePropertyConfig` in the `wsadmin` environment of the profile:

```
$AdminTask updatePropertyConfig {-serverName server_name -nodeName  
node_name -propertyFileName "profile_root\BusinessSpace\node_name\  
server_name\mm.runtime.prof\config\ConfigService.properties" -prefix  
"Mashups_"}
```

 and run `$AdminConfig save`.
3. Restart the server.

If you want to change the superuser when security is not enabled, perform the following steps on a cluster:

1. Modify the metric `noSecurityAdminInternalUserOnly` in the configuration file `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties`.
2. Run the command `updatePropertyConfig` in the `wsadmin` environment of the deployment environment profile:

```
$AdminTask updatePropertyConfig  
{-clusterName cluster_name -propertyFileName  
"deployment_manager_profile_root\BusinessSpace\cluster_name\  
mm.runtime.prof\config\ConfigService.properties" -prefix "Mashups_"}
```

 and run `$AdminConfig save`.
3. Restart the deployment manager.

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 administrative security is enforced.

Procedure

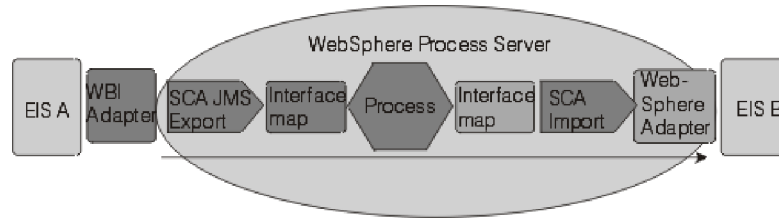
1. Determine which of the examples provided in this section most closely match your security needs. In some instances, your needs might involve a combination of information from more than one of the scenarios.
2. Read the security information for the relevant scenarios and apply it to your security needs.

Example

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 and a second interface

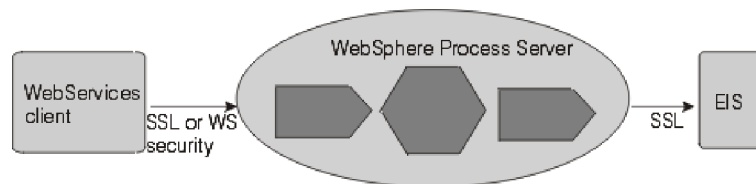
map and is then passed on to a second EIS (B), by way of a WebSphere Adapter. These are SCA invocations, with one component invoking a method on the next component.



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

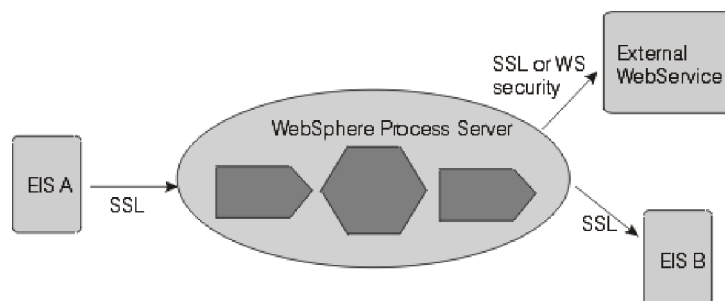
In this scenario, a Web service client invokes a WebSphere Process Server component. The request passes through several components in the WebSphere Process Server environment before being passed to an EIS by an adapter.



You can authenticate the Web service client as an 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

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



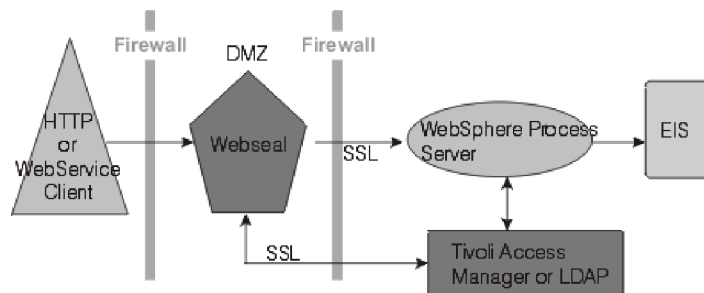
As for the inbound Web service request, you can authenticate with the external Web service as an SSL client, using HTTP Basic authentication or using WS-Security authentication. Use LTPACallbackHandler as the callback mechanism to extract the usernameToken from the current RunAs subject. Between WebSphere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





Printed in USA