CA Workload Automation

Security Guide

r11.3

CA Product References

This document references the following CA products:

- CA Access Control
- CA AutoSys Workload Automation Connect Option (CA AutoSys WA Connect Option)
- CA Embedded Entitlements Manager (CA EEM)
- CA Job Management Option
- CA Jobtrac™ Job Management (CA Jobtrac JM)
- CA Network and Systems Management (CA NSM)
- CA NSM Event Management
- CA NSM Management Command Center (CA NSM MCC)
- CA Scheduler® Job Management (CA Scheduler JM)
- CA Service Desk
- CA Universal Job Management Agent (CA UJMA)
- CA Workload Automation AE (formerly named CA AutoSys Workload Automation)
- CA Workload Automation Agent for UNIX (CA WA Agent for UNIX)
- CA Workload Automation Agent for Linux (CA WA Agent for Linux)
- CA Workload Automation Agent for Windows (CA WA Agent for Windows)
- CA Workload Automation Agent for i5/OS (CA WA Agent for i5/OS)
- CA Workload Automation Agent for Application Services (CA WA Agent for Application Services)
- CA Workload Automation Agent for Web Services (CA WA Agent for Web Services)
- CA Workload Automation Agent for Databases (CA WA Agent for Databases)
- CA Workload Automation Agent for SAP (CA WA Agent for SAP)
- CA Workload Automation Agent for PeopleSoft (CA WA Agent for PeopleSoft)
- CA Workload Automation Agent for Oracle E-Business Suite (CA WA Agent for Oracle E-Business Suite)
- CA Workload Automation Agent for z/OS (CA WA Agent for z/OS)
- CA Workload Automation EE (formerly named CA ESP Workload Automation)
- CA Workload Automation SE (formerly named CA 7 Workload Automation)

- CA Workload Control Center (CA WCC)
- CA Desktop and Server Management (CA DSM)

Contact CA

Contact Technical Support

For your convenience, CA provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA product documentation, complete our short customer survey, which is also available on the CA Support website, found at http://ca.com/docs.

Contents

Chapter 7: CA Workload Automation AE Policy Migration 153

Chapter 8: CA EEM Data Replication/Backup for CA WCC 175

Chapter 1: Introduction to Security

This section contains the following topics:

Overview

CA Workload Automation AE and CA WCC include features that let you secure objects such as jobs, calendars, cycles, global variables, machines, and resources, and delegate role-based access to them to the users in your enterprise. This guide explains these security features and helps you make decisions to properly secure your enterprise.

CA Workload Automation AE Security

CA Workload Automation AE includes features that let you secure objects such as jobs, calendars, cycles, global variables, machines, and resources. You can delegate administrative privileges to these objects to specific users or user groups.

CA Workload Automation AE provides security in the following ways:

- System-level security
- Native security
- External security

More information:

CA Workload Automation AE System-Level Security (see page 28)

Native Security Mode

Native security is the default security mode that CA Workload Automation AE runs under.

In native security mode, you can do the following to secure objects and delegate administrative privileges to users:

- Define EDIT superusers who have administrative privileges.
- Define EXEC superusers who have the authority to issue the sendevent command to send the execute events that affect the running of a job or the state of a job.
- Manage user IDs that jobs run under (job owners).
- Modify edit and execute permissions on a job-by-job basis.

Important! Although native security mode provides a level of security for certain objects and activities, the level of protection that native security mode provides is limited compared to that of external security. Only external security lets you control role-based access to objects (such as jobs, calendars, cycles, global variables, machines, and resources) at a granular level. Instead of using native security, we recommend that you enable external security after installation by configuring CA Workload Automation AE to work with CA EEM.

External Security Mode

External security is enabled by integrating CA Workload Automation AE with CA EEM. The external security mode is robust and provides better flexibility than the native security mode.

An EXEC superuser can enable external security by using the autosys_secure command. When external security mode is enabled, CA EEM is used to assign administrative rights to a user to define policies and to check whether a given user can switch the security mode of CA Workload Automation AE back to native. CA EEM lets you manage your user base, create roles for your enterprise, and assign roles to users. It also maintains security policies that govern what objects can be accessed by which users.

Note: While external security mode is enabled, native security is not enforced.

CA WCC Security

CA WCC uses CA EEM to secure objects and delegate access to its users. These objects are typically secured by the use of identities, resource classes, and policies.

You can use CA EEM to authenticate the CA WCC user and grant access for specified resources using a set of policies that are associated with a resource class.

More information:

CA WCC Identities (see page 85)
CA WCC Resource Classes (see page 96)
CA WCC Policies (see page 119)

Identities

Identities are the users and user groups that CA EEM employs to grant access to CA Workload Automation AE and CA WCC objects. The Manage Identities tab contains the pages that lets you list, search, view, and maintain global users, application-specific users, global user groups, and application-specific user groups.

Resource Classes

Resource classes are groups of resources that are used in CA EEM to control access to CA Workload Automation AE and CA WCC objects. You can create policies in each resource class to control user access to specified objects. For example, in CA Workload Automation AE, you can create policies for the as-job resource class to control access to jobs. In CA WCC, you can create policies for the ObjectAccess resource class to control access to specified jobs.

Note: In this usage, the term *resource* refers to jobs, calendars, cycles, global variables, machines, interface components, servers, and CA Workload Automation AE real and virtual resources.

More information:

Resource Class Details (see page 57)

Policies

A policy is a set of rules associated with users or user groups to define access to a particular object. Policies are created to do the following:

- Control access to objects such as jobs, calendars, cycles, machines, global variables, resources, and the owner field of a job.
- Prevent unauthorized users from starting or shutting down the scheduler or disabling external security.

Before performing an action on a specified object, CA Workload Automation AE issues a security call to the appropriate resource class in the repository. For example, for jobs, CA Workload Automation AE queries policies in the as-job resource class. For global variables, CA Workload Automation AE queries policies in the as-gvar resource class.

CA WCC has a single default policy for each resource class. Each policy is delivered with the CA WCC application in CA EEM.

Policy Types

Policies are classified according to their functions. You can create the following types of policies using CA EEM:

Access Policies

Defines the access rules for a particular CA Workload Automation AE object or CA WCC component.

CA Workload Automation AE provides a resource class for each object. You can create policies in each resource class to control the user access to the corresponding CA Workload Automation AE object. For example, for jobs, you must create policies in the as-job resource class. For global variables, you must create policies in the as-gvar resource class.

CA WCC provides a resource class for each component or group of components, and allows or denies identities (users or user groups) access to application resources. You can also define policies to be effective for a particular period by specifying a calendar.

Delegation Policies

Delegates users' authority to other users.

Dynamic User Group Policies

Defines application-specific groups and their memberships based on rules.

Event Policies

Determines which events are delivered and which ones are simply coalesced into summaries. By using event policies, you can configure which events are reported in detail.

Obligation Policies

Defines the application-specific rules that control what actions to perform when access is granted or denied. The obligation policies contain one or more obligation names and attributes. For example, the application may send an event, start a workflow process, or send an email.

Scoping Policies

Limits administrator access to CA EEM objects such as policies, calendars, and so on.

The three types from which you can select when creating policies are as follows:

- Access Policy—If the type is set to Access Policy, the actions and filters apply to all listed resources.
- Access Control List—If the type is set to Access Control List, each listed resource has its own actions and one filter or no filter.
- Identity Access Control List—If the type is set to Identity Access Control List, the following occur:
  - Actions are set to particular identities. A default rule that applies to all identities not in the list is created.
  - Identity types are marked with icons (users, application groups, global groups, or dynamic groups).

  There is a simple list for the resources. Filters do not exist for this type of policy.

Policy Permissions

All policies assigned to the identities have grant and denial rights, as follows:

Explicit Grants

Provides the identities with the access rights specified to the resources specified when the policy evaluates true. Most policies you create are explicit grants.

Explicit Denies

Prevents the identities from the access rights specified to the resources specified when the policy evaluates true. You can use an explicit deny policy if you need to deny access to a single user within a group. For example, perhaps you have created explicit grant policies for a group, but one member of the group should not have permissions to issue the sendevent command. You can create an explicit deny policy in CA Workload Automation AE using as-control resource class and assign that policy to that user only. You can also create the same explicit deny policy in CA WCC using the AccessControl resource class and assigning that policy to that user only. All other access rights granted to the group will still apply.

Policy Evaluation

During an authorization check, policy evaluation occurs in two phases—the match phase and the evaluation phase.

During the match phase, all policies for the specified resource class are matched against the following fields: policy's identities, resource name, actions, and calendar. If any of these are empty, they are assumed to include all possible values.

During the evaluation phase, matching policies are further evaluated against their respective filters. Each filter contains the following fields: Logic, Left type/value, Operator, and Right type/value, with parentheses surrounding the sub-expression. Filters are similar to the where clause in a database query. You can define multiple filters by combining them using the AND and OR operators. If a matched policy has no filters, or if the policy filters evaluate to true, the policy is assumed to have granted access.

Example: Using a filter to limit server access to a specified group

Suppose you have servers located in Florida where all of the server names begin with FL, and you want to create an access policy that limits their use to only those users located in Florida. To do this, you will create a ServerAccess policy where the resource is servers/Fl*, and then add a filter as follows:

```
Where: global user: State == value: FL
```

When the policy is evaluated, only the users living in Florida will be able to access the Florida servers.

Best Match Policy Evaluation

When deciding whether to grant access to an object, CA EEM only considers policies whose resource names most closely match the requested object name. The policy may contain the most matching characters and the fewest wildcard characters compared to the object name given to it by CA Workload Automation AE or CA WCC. To evaluate whether a user has access to an object, only policies that CA EEM has collected using the best match criteria are considered. Consequently, the best match policy evaluation used by CA EEM lets you benefit from a hierarchical object naming convention.

Example: Best Match Policy

This example lets you select a job naming convention where all payroll jobs in the ACE instance are prefixed with the payroll identifier. Using such a job naming convention lets you create a generic policy that can govern all payroll jobs, where the resource name in the policy is ACE.payroll*. At the same time, you can fine-tune security for a specific job by creating a policy for a job named payroll_employeeID5702, where the resource name in the policy is ACE.payroll_employeeID5702.

With best match policy evaluation, CA EEM only considers the policy containing the resource name ACE.payroll_employeeID5702 when deciding whether to grant access to the payroll_employeeID5702 job. Even though there may be other matching policies, such as the policy with a resource name of payroll*, CA EEM uses the best match policy evaluation to find the set of policies with resource names that most closely match the requested object name.

Filtering and Regular Expressions

You can reduce the overall number of security policies you need to create by using the filter functionality provided in CA EEM. Instead of creating multiple policies, you can add filters to a single policy to control access to resources. Filters use the regular expression syntax.

A regular expression is a special text string used to describe a search pattern and is used for creating filtering criteria in CA EEM policies. Having a good understanding of how to use regular expressions in CA EEM policies is essential when creating filters to limit the scope of your policies.

Services

The following services must be installed and running on the CA EEM r8.4 server:

Windows

- CA Directory - iTechPoz-*hostname*—Embedded CA Directory DSA (directory service agent) for the CA EEM LDAP repository.
- CA Directory - iTechPoz-*hostname*-Router—Embedded CA Directory router DSA (directory service agent) for the CA EEM LDAP repository that handles LDAP requests from other CA EEM components.
- CA Directory SSL daemon - iTechPoz-Server—Embedded CA Directory SSL daemon that handles LDAP authentication/encryption requests over SSL for the CA EEM LDAP repository.
- CA iTechnology iGateway 4.5— iTechnology iGateway service that handles the requests coming from CA EEM clients, processes the requests, and sends responses to the clients.

UNIX

- For CA Directory:
  - dxserver start iTechPoz-*hostname*
  - dxserver start iTechPoz-*hostname*-Router
  - ssld start iTechPoz-Server
- For iGateway:
  - ./igateway –b
  - /bin/sh ./WDigateway.sh

Chapter 2: Best Practices for Setting Up Security

To secure workload automation objects, you can use native security on CA Workload Automation AE with CA EEM policies on CA WCC, or you can define CA EEM policies on both CA Workload Automation AE and CA WCC. We strongly recommend you implement CA EEM security on both. Either way, you can follow the best practices described in this section when setting up security.

This section contains the following topics:

CA Workload Automation AE Security Mode

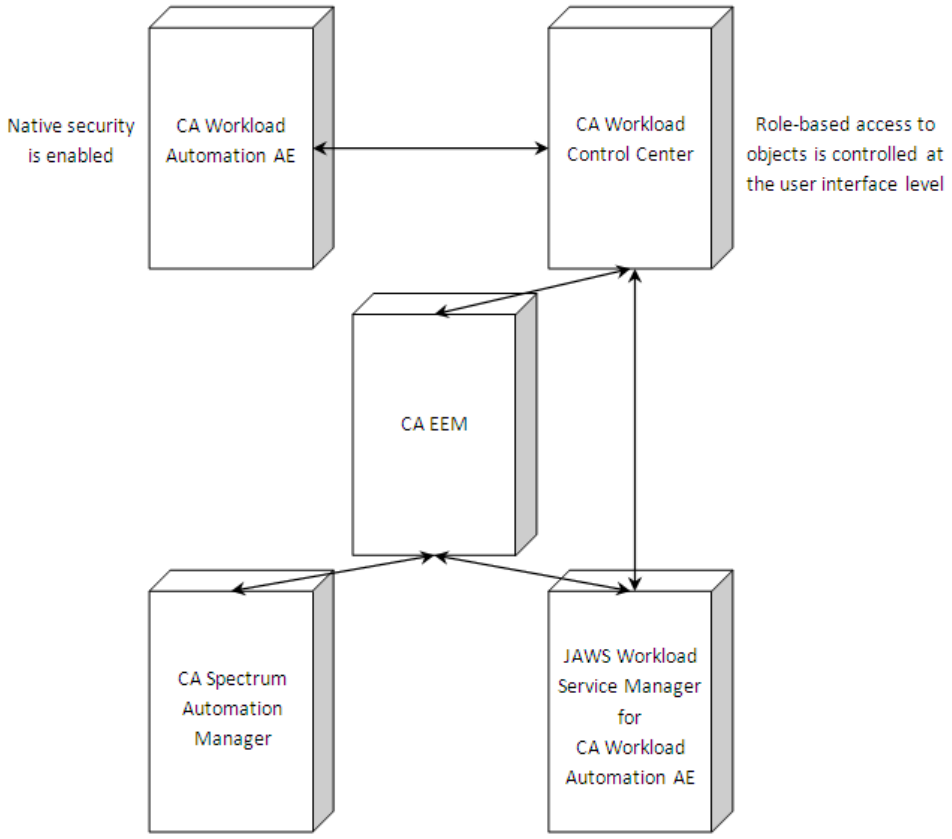
By default, CA Workload Automation AE uses native security and system-level security. However, the level of protection that native security mode provides is limited compared to that of external security (integration with CA EEM).

Important! As a best practice, we recommend that you enable external security after installation by configuring CA Workload Automation AE to work with CA EEM. Only external security lets you control role-based access to objects (such as jobs, calendars, cycles, global variables, machines, and resources) at a granular level.

However, if you must use native security mode on CA Workload Automation AE, we recommend the following:

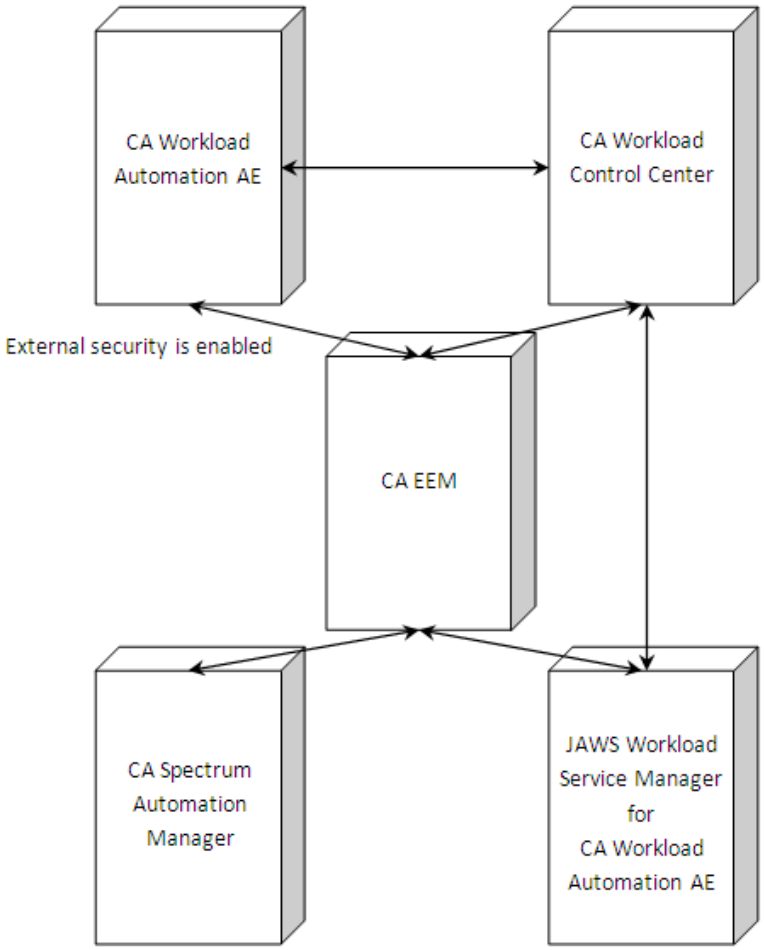
- Create CA EEM policies on CA WCC that control what objects users can see and modify when they use CA WCC. You must ensure that these policies cover all the objects stored on CA Workload Automation AE that you want to secure. This is because granular, role-based security is only enforced at the user interface level in this environment.
- Remove icons and options from CA WCC that users do not need. By removing options from the user interface, the users in your environment cannot try to access features that they do not need.
- Specify the owner and permission attributes in the job definitions that are created using the jil command on CA Workload Automation AE. CA EEM policies defined on CA WCC does not check security for objects defined using the jil command, so you can use the ower and permission attributes grant the appropriate permissions to users. You must also define appropriate operating system user IDs and passwords in the database that restrict who can log on to and run jobs on client computers.

The following diagram shows the configuration where CA Workload Automation AE uses native security mode and CA WCC uses CA EEM on a dedicated server:



Recommended Security Configuration

To secure your environment, we recommend that you configure CA Workload Automation AE and CA WCC to use CA EEM as shown in the following diagram:



In this configuration, CA EEM is a common component used by multiple products, including CA Workload Automation AE and CA WCC.

Notes:

- This recommended configuration shows CA Workload Automation AE and CA WCC using the same CA EEM instance that is installed on a dedicated server. Alternatively, you can configure CA Workload Automation AE to work with an existing CA EEM instance that is installed on the CA WCC server. If CA EEM and CA WCC are installed on the server, ensure that the server has sufficient memory to handle the processing for both components. We also recommend that you use a multi-CPU or a high performance CPU.
- To avoid a single point of failure and to avoid potential performance issues, we do *not* recommend installing CA EEM and CA Workload Automation AE on the same server. Most security policies are typically created and checked on the CA Workload Automation AE server.

Policy Differences

The CA EEM policies for CA Workload Automation AE are categorized by object type (for example, jobs, calendars, and resources). CA Workload Automation AE provides a resource class for each object type (for example, as-job, as-calendar, and as-resource). You can define a policy in the appropriate resource class that controls all access modes (read, create, modify, delete, execute) for a particular object or group of objects.

In contrast, CA EEM policies for CA WCC are classified according to their functions. For example, the ObjectAccess resource class controls which objects a user can view in CA WCC (read access). The ObjectControl resource class controls the actions that can be performed on a particular class of objects (create, modify, and delete access).

Note: However, an ObjectControl policy does not control access to individual objects. For example, a user must be authorized to create a job.

Users and user groups are associated with these policies to restrict the objects that the users have access to.

We recommend the best practices in this section so that you can coordinate the policy designs between CA Workload Automation AE and CA WCC to avoid duplicate policy checks and achieve the best possible performance.

Avoid Redundant Policies on CA Workload Automation AE and CA WCC

Important! If you create policies on both CA Workload Automation AE and CA WCC that define the same access control, your environment may experience performance issues because the security checks are done twice. For example, when updating data stored on the CA Workload Automation AE event server (such as resource and job definitions), a CA WCC policy may check permissions for accessing an object in the user interface and a CA Workload Automation AE policy may check permissions for accessing that same object on the database.

You can avoid redundant policies for data that is stored on the CA Workload Automation AE event server, such as job and resource definitions. We recommend the following:

- Ensure that the Filter Object setting is disabled (the default) on the CA Workload Automation AE Server Properties page in CA WCC Configuration Manager for your server.
- Define policies that control the access granted to specific objects or groups of objects on CA Workload Automation AE only.

With this setup, CA WCC does not check for CA EEM policies when retrieving objects from CA Workload Automation AE. The CA EEM security checks are only done on CA Workload Automation AE. In other words, CA WCC lets users try to perform tasks (such as access a job or calendar) at the user interface level, but CA Workload Automation AE determines whether they have the permissions to access or perform those tasks.

Because the security checks are done on the scheduling manager, this setup also protects data when you use the CA Workload Automation AE commands (such as jil) instead of CA WCC to control, configure, and report jobs.

Notes:

- You cannot avoid redundant policies for data that is copied from CA Workload Automation AE and stored locally on the CA WCC server, such as monitoring data. For this type of data, you must ensure that the CA WCC policies match the CA Workload Automation AE policies.
- You might want to define security policies on CA WCC to restrict permissions by user role at the user interface level. For example, suppose that your scheduling environment is used by three groups of external users (group A, group B, and group C). The user IDs are defined on external networks that you do not have access to. To let these groups access job objects in your environment, you give these users generic user IDs to log into CA WCC. To restrict group A's access to their jobs only, you can define security policies on CA WCC that restrict access by their generic user IDs.

Use a Naming Convention for Objects and Policies

When defining objects (such as jobs, calendars, cycles, global variables, machines, and resources), we recommend that you use a naming convention that groups objects by categories. For example, you can prefix the names of all payroll jobs with **payroll_**, and you can prefix the names of all renewable resources with **renewable_**.

By using a naming convention for objects, you can coordinate the security configurations in CA Workload Automation AE, CA WCC, and CA EEM as follows:

- You can define CA EEM policies that apply to groups of related objects that share similar security restrictions.
- You can help ensure that the correct objects are secured as determined by the best match policy evaluation that CA EEM uses.

Examples of how the naming conventions are used are described in the following topics.

More Information:

Use a Naming Convention for the Resources Specified in Policies (see page 23)
Use a Naming Convention for Policy Names (see page 24)

Use a Naming Convention for the Resources Specified in Policies

When you create a policy in CA EEM, the following conventions apply when naming the resources (objects) that you want to secure:

CA Product	Resource Class	Naming Convention for the Resources (Objects) Specified in the Policy
CA Workload Automation AE	All resource classes except as-owner	*AEinstance.objectName*
CA WCC	ObjectControl	server/*serverName*/*objectType* server/*serverName*/Job/*jobType*
CA WCC	ObjectAccess	server/*serverName*/*objectType*/*objectName*

In all these conventions, *objectName* specifies the name of the object that you want to secure.

If you use a standard naming convention for your object names, you can use wildcards in the *objectName* value to create CA EEM policies that restrict access to groups of objects.

Also, CA EEM uses a best match policy evaluation to determine whether to grant access to objects. CA EEM only considers policies whose resource names most closely match the requested object name. Therefore, by following a standard naming convention for your object names, you can help ensure that the correct objects are matched.

Note: If you have not used a standard naming convention for your existing objects, we recommend that you start using a naming convention for new objects.

Use a Naming Convention for Policy Names

To coordinate policies between CA WCC and CA Workload Automation AE, we recommend that you use a standard naming convention for policy names. By using a naming convention that describes the intent of the policies, the policies are easier to understand and manage.

For example, the following CA WCC and CA Workload Automation AE policies use the *product-roles-accessType-objectType* naming convention:

```
WCC-SCHEDOPER-CONTROL-ALLJOBS
WCC-SCHEDOPER-ACCESS-ALLJOBS
AE-SCHED-ACCESSCONTROL-PAYROLLJOBS
AE-OPER-ACCESSCONTROL-PAYROLLJOBS
```

The names indicate that a CA WCC ObjectControl policy and a CA WCC ObjectAccess policy specify the object permissions for the scheduler and operator roles for all jobs. On CA Workload Automation AE, an as-job policy specifies the permissions for the scheduler role for payroll jobs. Another as-job policy specifies the permissions for the operator role for payroll jobs.

Example: Define Policies for Payroll Jobs

This example shows policies that follow the security best practices. Suppose that your enterprise runs jobs associated with payroll in the CA Workload Automation AE instance ACE. CA EEM policies that govern who can access these jobs are created using the following guidelines:

- Avoid redundant policies.

  The CA WCC policies are general and are categorized by user roles. The CA Workload Automation AE policies are specific and define the access modes for the objects.

- Use a standard naming policy for objects.

  The names of all the payroll jobs are prefixed with **payroll_**. This naming convention lets you define CA EEM policies that protect these specific objects. The following are sample job names:

  - payroll_employeeID5701
  - payroll_employeeID5702
  - payroll_employeeID5703
  - payroll_employeeID6002
  - payroll_employeeID6003

- Use a standard naming convention for the resources specified in the policies.

  The resources (job objects) specified in the CA WCC policies use the following naming standard:

  server/*serverName*/Job/*jobType*

  server/*serverName*/JOB/*object*

  The resources specified in the CA Workload Automation AE policies use the following naming standard:

  AEinstance.objectName

- Use a standard naming convention for policy names.

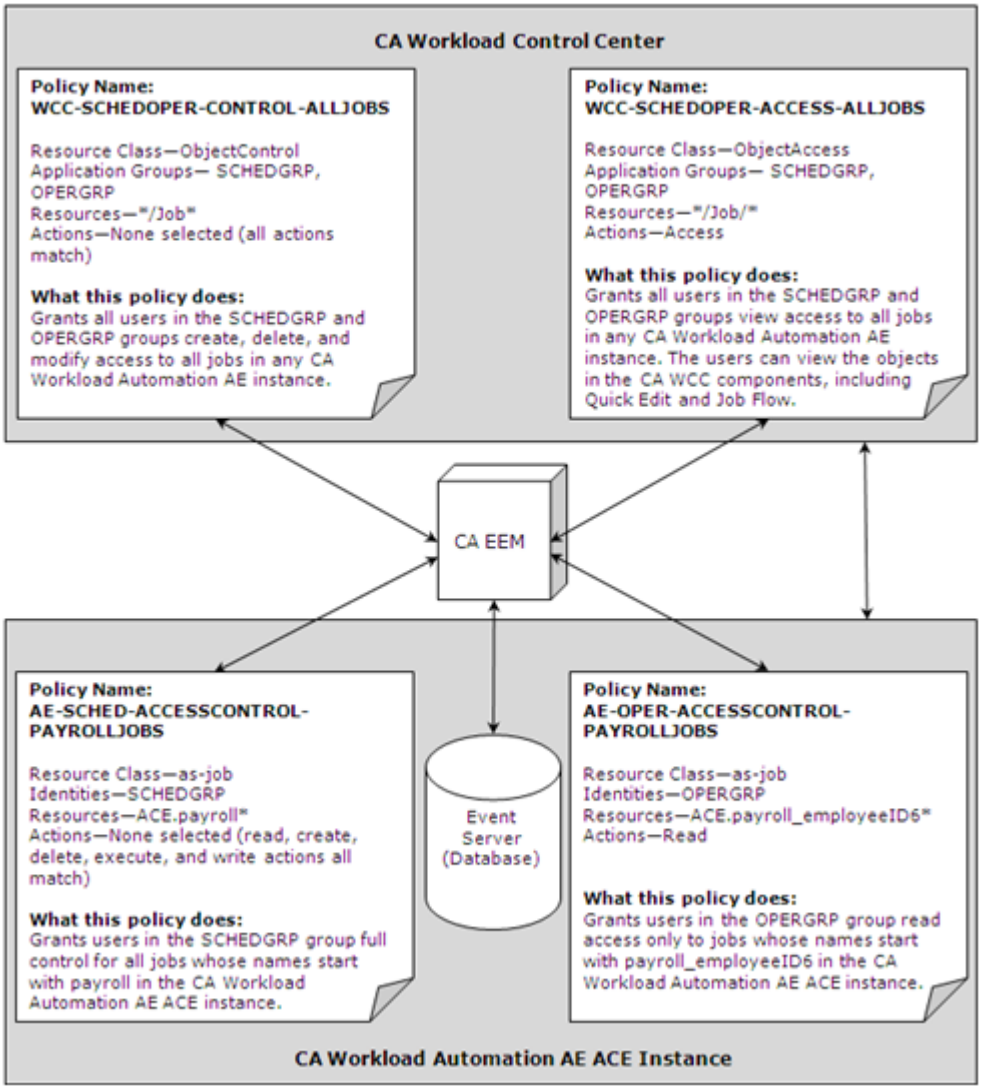
  To coordinate the policies among CA Workload Automation AE, CA WCC, and CA EEM, the following naming convention is used for the policy names:

  product-roles-accessType-objectType

The following application groups (user groups) are defined in CA EEM:

- SCHEDGRP—The users in this group need full control (read, create, delete, execute, and write) to all payroll jobs in the CA Workload Automation AE instance ACE. This user group includes the default CA WCC ejmscheduler, ejmexec, and ejmsupervisor users.
- OPERGRP—The users in this group are primarily responsible for creating monitors. They only need read access to payroll jobs whose names start with **payroll_employeeID6** in the CA Workload Automation AE instance ACE. This user group includes the default CA WCC ejmoperator user.

The following diagram describes the policies that are created:



Chapter 3: Authentication and Authorization

The three key objects of CA EEM are identities (users and user groups), resource classes, and policies. CA EEM is used to secure CA WCC objects, providing the following capabilities:

Authentication

CA EEM authenticates the user. The authenticated user can then be used in subsequent authorization processing.

Authorization

CA EEM provides access to a user for a particular resource. A resource can be any logical or physical entity. In CA Workload Automation AE, the typical resource is a job, calendar, cycle, global variable, real or virtual machine, and so on. In CA WCC, the typical resource is a user interface component (for example, tab, command, drop down list, and so on). Authorization is controlled by a set of policies associated with a resource class. These policies are the primary way to integrate CA EEM with CA Workload Automation AE and CA WCC.

This section contains the following topics:

CA Workload Automation AE System-Level Security

System-level security prevents the following:

- Unauthorized access to job information.
- Unauthorized jobs from running on a machine.

CA Workload Automation AE provides the following security features at the system level:

- Database field verification
- Job definition encryption

  Note: This applies to legacy agents only.
- Remote authentication
- User and database administrator passwords
- File system access restriction
- Data encryption

These security features are always available and are in effect regardless of the active security mode.

Note: On UNIX, the database field and control string encryption features provide a level of security comparable to the security provided in the native UNIX environment.

Database Field Verification

To secure the database, CA Workload Automation AE does the following:

1. Encrypts some of the fields specified in a job definition.
2. Generates a checksum from the fields in the job definition.
3. Stores the checksum in the database.

When a job is accessed, its checksum is regenerated and compared to the one in the database. If the checksums are different, it indicates that the job definition in the database has been modified, probably by using an SQL command. In this case, the job is disabled and cannot be run.

To enable a disabled job, edit the job definition using the update_job subcommand or CA WCC, and save it.

Note: You must be the owner of the job or the EDIT superuser to edit the job definition.

Job Definition Encryption

To secure the legacy agents from unauthorized access, the scheduler encrypts the information in a job definition that is sent to the legacy agent. The legacy agent decrypts the job information and processes the job. If the legacy agent receives any job information from the scheduler that it does not recognize, it issues an error message and the job is not processed.

Remote Authentication

CA Workload Automation AE uses the following remote authentication methods to authenticate an agent before permitting it to run a job on a computer:

- User authentication—Verifies whether a user has permission to start a job on a client.
- Scheduler authentication—Verifies whether a scheduler has permission to start a job on a client.

  Note: Scheduler authentication applies to legacy agents only.

The remote authentication methods are stored in the database and referenced when a client initializes. By default, both user authentication and scheduler authentication are disabled. The EDIT superuser must use the autosys_secure command to enable them. If you enable scheduler authentication, you must configure CA Workload Automation AE to support it.

More information:

Configure Scheduler Authentication for Legacy Agents on UNIX (see page 33)

User Authentication

CA Workload Automation AE uses the user authentication method to verify whether a user has permission to start a job on a client. By default, user authentication is disabled. Only the EDIT superuser can enable it using the autosys_secure command.

UNIX:

User authentication uses the UNIX ruserok() function to verify whether a user has permission to start a job on a client computer. The client's agent makes the ruserok() UNIX system call. The ruserok() UNIX system call checks the client computer's /etc/hosts.equiv file and the user's .rhosts file to validate that the requesting user is registered in that environment. This function call performs a local verification and is not related to rshd or rlogind.

The hosts.equiv or .rhosts file entries must match the job owner and machine name values exactly. For example, if the job owner is parrot@jungle, the hosts.equiv or .rhosts file must contain jungle. Similarly, if the owner is parrot@jungle.vine.com, the hosts.equiv or .rhosts file must contain jungle.vine.com. If the values do not match, jobs fail to run on that computer when ruserok() remote authentication is used.

Note: For more information about ruserok(), see the documentation for your UNIX system.

Windows:

User authentication uses Microsoft authorization technologies to verify whether a user has permission to start a job on a client computer. The primary domain controller is obtained and contacted to validate that the requesting user is registered in that environment.

How Scheduler Authentication Works

Note: Scheduler authentication applies to legacy agents only.

CA Workload Automation AE uses the scheduler authentication method to verify whether a scheduler has permission to start a job on a client. By default, scheduler authentication is disabled. Only the EDIT superuser can enable it using the autosys_secure command.

When scheduler authentication is enabled, the legacy agent verifies whether it has permissions to process requests from the requesting scheduler before starting a job.

Scheduler authentication works as follows:

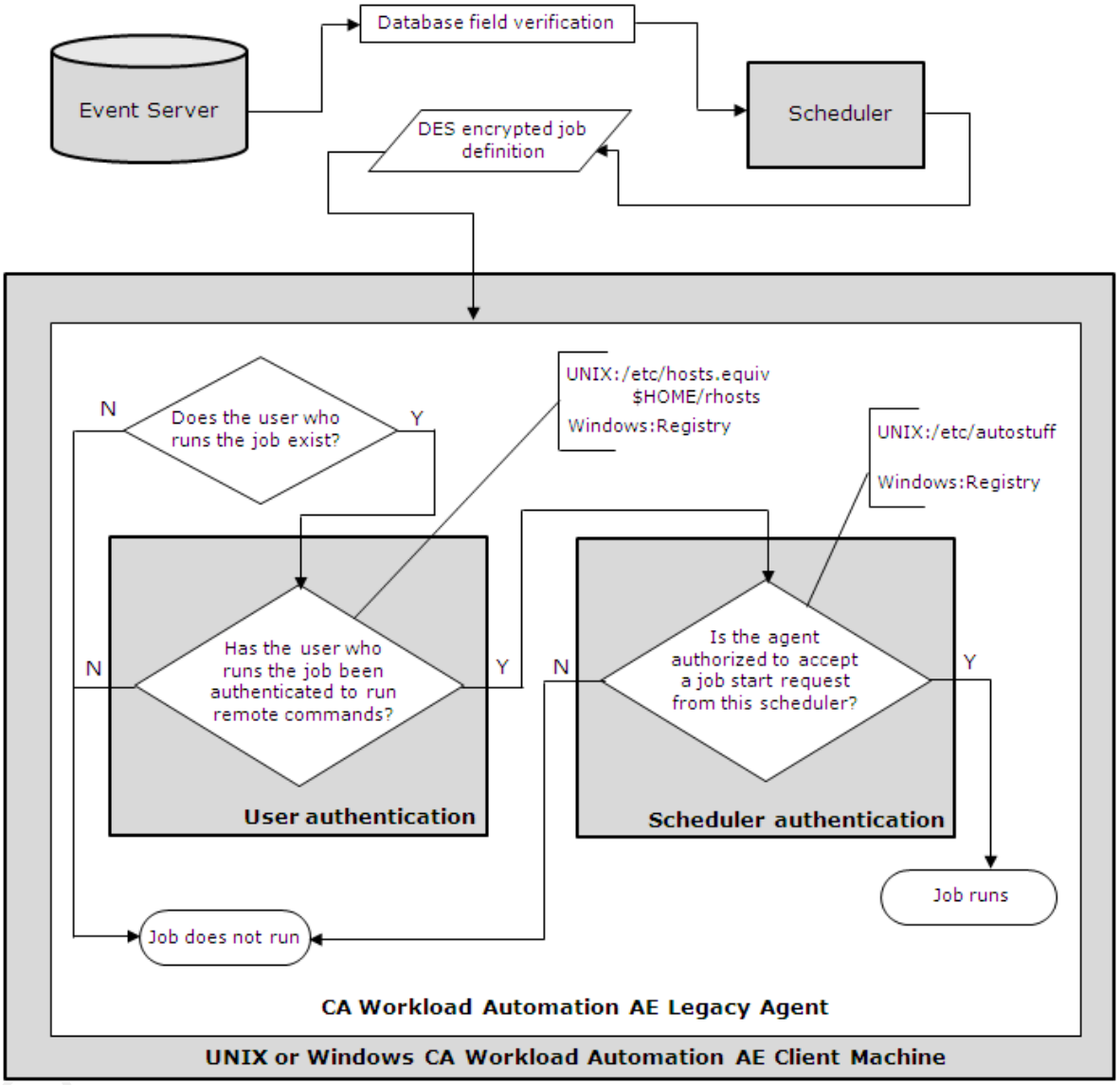
- UNIX—The /etc/.autostuff file is read. This file is located on the computer where the legacy agent is running. Before enabling scheduler authentication, you must set up and configure the /etc/.autostuff file on every client computer that requires scheduler authentication.
- Windows—The Authorized Scheduler Host Names registry entry is read. This registry entry is located on the computer where the legacy agent is running. Before enabling scheduler authentication, you must set up and configure the Authorized Scheduler Host Names registry entry on every Windows client computer that requires scheduler authentication. This registry entry is configured using the Unicenter AutoSys Agent window of the Administrator (autosysadmin) utility.

  Note: For more information about the Administrator utility, see the *Online Help*.

The scheduler checks whether the following starting conditions are satisfied before running a job on an agent computer:

- Has the job definition been modified? If so, the job definition is invalid and the job does not run.
- Can the scheduler connect to the agent computer as defined in the DES-encrypted job definition?
- Does the user defined as the job owner (user@machine) have a logon account on the agent computer?
- If user authentication is enabled, the following conditions are checked:
  - On UNIX, is the user a trusted user as defined in the /etc/hosts.equiv and $HOME/.rhosts files?
  - On Windows, is the user a trusted user as verified by the primary domain controller machine?
- If scheduler authentication is enabled, does the requesting scheduler have permission to run jobs on the agent computer?

The following illustration shows the permissions and security checks that occur before a job is allowed to start:



Note: In the illustration, an asterisk (for example, Y*) indicates checks that are made only if the specific method of remote authentication is enabled.

Configure Scheduler Authentication for Legacy Agents on UNIX

You can configure the legacy agents to accept jobs only from authorized schedulers.

Note: The scheduler authentication applies to legacy agents only.

To configure scheduler authentication for legacy agents on UNIX

1. Log on to CA Workload Automation AE as the EDIT superuser and run the shell that is sourced to use CA Workload Automation AE.
2. Enter the following command at the operating system prompt:

   ```
   autosys_secure
   ```

   The following menu appears:

   ```
   Please select from the following options:
   [1] Activate EEM instance security.
   [2] Manage EDIT/EXEC superusers.
   [3] Change database password.
   [4] Change remote authentication method.
   [5] Manage user@host users.
   [6] Get Encrypted Password.
   [0] Exit CA WAAE Security Utility.
   ```

3. Enter 4 and press the Enter key.

   The following menu appears:

   ```
   Please select from the following options:
   [1] No remote authentication.
   [2] Agent user authentication only.
   [3] Scheduler authentication only.
   [4] Both Agent and Scheduler authentication ([2] & [3]).
   [9] Exit from "Change remote authentication method" menu.
   [0] Exit CA WAAE Security Utility.
   ```

4. Enter 3 and press the Enter key.

   The scheduler authentication is enabled.

5. Do the following for every client computer that requires scheduler authentication:

   a. Create an ASCII file named.autostuff in the /etc directory.

   b. Grant read and write permissions for the /etc/.autostuff file for root only.

   c. Define the following entry in the /etc/.autostuff file:

      AUTOSERV:hostname

      AUTOSERV

      Specifies a three-letter instance name associated with an authorized scheduler.

      hostname

      Specifies the host name associated with an authorized scheduler.

   CA Workload Automation AE is configured for scheduler authentication. The legacy agent now reads the /etc/.autostuff file to verify its permission to process a scheduler's requests before starting each job. The legacy agent only runs jobs submitted by the schedulers listed in the /etc/.autostuff file.

User and Database Administrator Passwords

When you install CA Workload Automation AE and configure the database, a database user named autosys is added and a database password is defined. The autosys user is granted rights to the CA Workload Automation AE objects and can make changes to specific information in the database. To enhance system security, we recommend that you use the autosys_secure command to change the autosys user password.

You must provide the autosys and sa (system administrator) user IDs and passwords to use specific utilities. For example, when using the ISQL utility to query the database, you must provide both the autosys user password and the sa password.

Notes:

- Only the EDIT superuser can use the autosys_secure command to change the autosys user password. For information about the autosys_secure command, see the *Reference Guide.*
- Every event server in an instance must have the same database password. If you are running in dual event server mode, the autosys_secure command changes the password on both event servers. If CA Workload Automation AE has rolled over to single event server mode, do not change the password until you have re-established dual event server mode.

Add a User ID and Password for a Legacy Agent Computer

To run jobs on a legacy agent computer, you must define user IDs and passwords that the jobs will run under. The user IDs are specified in job definitions using the machine attribute.

Notes:

- Before you can define user IDs and passwords, you must define the agent computer to CA Workload Automation AE using the JIL insert_machine subcommand. For more information about defining agent computers, see the *Reference Guide.*
- On UNIX, the user ID you enter in the owner attribute has the authority to run the job on the agent computer. The user default shell is used.

To add a user ID and password for a legacy agent computer

1. Log on to CA Workload Automation AE as the EDIT superuser and enter the following command at the UNIX operating system prompt or the Windows command prompt:

   ```
   autosys_secure
   ```

   The following menu appears:

   ```
   Please select from the following options:
   [1] Activate EEM instance security.
   [2] Manage EDIT/EXEC superusers.
   [3] Change database password.
   [4] Change remote authentication method.
   [5] Manage user@host users.
   [6] Get Encrypted Password.
   [0] Exit CA WAAE Security Utility.
   ```

2. Enter 5 and press the Enter key.

   The following menu appears:

   ```
   Please select from the following options:
   [1] Create user@host or Domain password.
   [2] Change user@host or Domain password.
   [3] Delete user@host or Domain password.
   [4] Show all user@host users.
   [9] Exit from "Manage user@host users" menu.
   [0] Exit CA WAAE Security Utility.
   ```

3. Enter 1 and press the Enter key.
4. Enter the user name, user host or domain, and password information when prompted.

   The user is added. The following message appears:

   ```
   CAUAJM_I_60135 User Create successful.
   ```

Example: Specify a User ID and Legacy Agent Computer in a Job Definition

This example runs a command on a legacy agent computer named legacyagenthost. The command runs under the sched user. This user is the job owner and is specified using the owner attribute.

```
insert_job: aslegacyjob
job_type: CMD
owner: sched@legacyagenthost
machine: legacyagenthost
command: my_command
```

The owner must have an account on the target legacy agent computer. You must specify the owner of the job definition as *user@machine*. The account must match the owner name exactly for the job to run.

File System Access Restriction

You can prevent unauthorized use of CA Workload Automation AE by restricting access at the file system level as follows:

- Make sure that only authorized users can change permissions on the files and directories in the directory structure.
- Check the level of security you must use. For example, you can restrict access as follows:
  - Only authorized users can use CA Workload Automation AE.
  - Any user can view reports about jobs. For example, you can use the autorep command to view the status of a job, but only authorized users can create jobs and calendars or make changes to them.

If you want only authorized users to access CA Workload Automation AE, make sure that only those users have execute permissions for the files in the $AUTOSYS/bin directory.

If you want all users to view reports about jobs, but only authorized users to create and edit jobs and calendars, make sure that only the authorized users have execute permission for the following files in the $AUTOSYS/bin directory:

- autocal_asc
- archive_events
- autotimezone
- clean_files
- DBMaint
- dbspace
- dbstatistics
- jil
- sendevent

Restricting execute permission for these files prevents unauthorized users from making configuration changes.

You must also protect the files in the $AUTOUSER directory from modification by ensuring that only users authorized to make configuration changes have write permission for the files. Read permission is necessary to source the environment files.

Data Encryption

CA Workload Automation AE supports the encryption of data and messages shared between the command line utilities, agent, scheduler, and the application server. CA Workload Automation AE uses the Advanced Encryption Standard (AES) algorithm to encrypt and decrypt data. This algorithm requires an encryption key to encrypt data.

CA Workload Automation AE encrypts data in the following communication scenarios:

- Application server and client utilities—The data exchanged between the command line utilities and the application server is encrypted using an instance-wide encryption key. This key is specific to an instance and must be the same on all computers where the server and clients are installed. During the CA Workload Automation AE installation, a default instance-wide encryption key is created and stored in the $AUTOUSER/cryptkey.txt file. However, you can define a user-specific encryption key using the as_config command or using CA Workload Automation AE Administrator on Windows.

  Note: For more information about the as_config command, see the *Reference Guide*. For more information about CA Workload Automation AE Administrator, see the *Online Help*.

- Application server and agent or scheduler and agent—The data exchanged between the application server and the agent or the scheduler and the agent is encrypted based on the encryption type and the encryption key specified in the machine definition and the agent. On CA Workload Automation AE, you can set the encryption type and encryption key to be used for each agent using the encryption_type and key_to_agent JIL attributes. The encryption key specified on CA Workload Automation AE must match the encryption key specified on the agent.

  Note: For more information about the encryption_type and key_to_agent JIL attributes, see the *Reference Guide.* For more information about setting up encryption on the agent, see the *CA Workload Automation Agent for UNIX, Linux, or Windows Implementation Guide*.

Considerations when Enabling Instance-Wide Encryption

The following are important considerations when you enable instance-wide encryption:

- The CA Workload Automation AE clients, agents, or the SDK operate only in encryption-enabled mode.
- If instance-wide encryption is enabled, communication between a legacy client and the CA Workload Automation AE server, a CA Workload Automation AE client and the legacy server, and the CA Workload Automation AE server and a legacy agent is disabled. No jobs are sent to legacy agents while the instance-wide encryption is enabled.
- You can decrypt data from an external instance or encrypt data to an external instance using the local instance-wide encryption key. Both the local and external instances must use the same instance-wide encryption key. In either of these instances, if you specify an encryption key other than the default key that is created during the installation, you must define the local instance-wide encryption key in the external instance definition. Otherwise, the request or response to that external instance is not encrypted.
- You must install eTrust Public Key Infrastructure (ETPKI). CA Workload Automation AE installs ETPKI when any of the following components are selected during the CA Workload Automation AE installation:
  - Server
  - Agent
  - Client

Set Instance-Wide Encryption on UNIX

You can set an instance-specific encryption key for all communication between the CA Workload Automation AE components of the same instance. This encryption key is stored in the $AUTOUSER/cryptkey.txt file. This key must be the same on all computers where the server and clients are installed for a particular CA Workload Automation AE instance.

During the CA Workload Automation AE installation, a default encryption key is created. However, you can modify this encryption key after installation.

To set instance-wide encryption on UNIX

1. Run the shell that is sourced to use CA Workload Automation AE.
2. Enter the following commands at the operating system prompt:

   ```
   unisrvcntr stop waae_sched.$AUTOSERV
   unisrvcntr stop waae_server.$AUTOSERV
   ```

   The scheduler and the application server stop.

3. Edit the following parameter in the configuration file, and save the file:

   ```
   UseEncryption=0|1|2
   ```

   0

   Specifies that no encryption is used.

   1

   Specifies that the default encryption key is used to encrypt data. This key is created during the CA Workload Automation AE installation.

   2

   Specifies that a user-specified encryption key is used to encrypt data.

   Note: If you set the UseEncryption parameter to 2, you must specify the encryption key using the as_config command.

4. Enter the following command at the operating system prompt:

   ```
   unisrvcntr start waae_sched.$AUTOSERV
   unisrvcntr start waae_server.$AUTOSERV
   ```

   The scheduler and the application server start. The instance-wide encryption is set.

Notes:

- For information about specifying the encryption key using the as_config command, see the *Reference Guide*.
- On Windows, you can select the equivalent value using the Use Instance Wide AES 128-bit Data Encryption check box on the Instance - CA Workload Automation AE Administrator window of CA Workload Automation AE Administrator. For information about setting instance-wide encryption on Windows, see the *Online Help*.

Agent Encryption

The data exchanged between the application server and the agent or the scheduler and the agent is encrypted based on the encryption type and the encryption key specified in the machine definition. On CA Workload Automation AE, you can set the encryption type and the encryption key to be used for each agent using the encryption_type and key_to_agent JIL attributes. On the agent, the encryption key is stored in the cryptkey.txt file located in the agent installation directory. The key specified on CA Workload Automation AE must match the key specified on the agent.

Notes:

- The agent encryption and instance-wide encryption are mutually exclusive. For example, you can disable instance-wide encryption and set the agent encryption type to either the default or AES. All combinations of encryption are supported.
- An instance can communicate with agents using different encryption settings.
- You can copy the cryptkey.txt file that is generated on one computer to another, provided the encryption type and the encryption key are the same.
- The encryption type and the encryption key must be the same for all agents defined with the same name. If you want to define the agents with different encryption types or encryption keys, you must define the agents with different names. You can define the agent name using the agentname parameter in the agentparm.txt file.
- For more information about the cryptkey.txt file and the agentparm.txt file, see the *CA Workload Automation Agent for UNIX, Linux, or Windows Implementation Guide*.
- For information about the encryption_type and key_to_agent JIL attributes, see the *Reference Guide*.

CA Workload Automation EE External Instance and CA WA Agent for z/OS Encryption

CA Workload Automation EE external instance definitions and the CA WA Agent for z/OS do not support encryption. However, because CA Workload Automation AE uses instance-wide encryption, you must configure the scheduler for CA Workload Automation AE to communicate with CA Workload Automation EE external instances or work with the CA WA Agent for z/OS.

Note: For more information about how to configure CA Workload Automation AE to work with CA WA Agent for z/OS or configuring cross-instance scheduling, see the *UNIX Implementation Guide* or *Windows Implementation Guide*.

Security Levels

CA WCC security is implemented on two levels, by the user session and through the back-end workload automation server.

Session Security

User session-level security is implemented through the Apache Tomcat session for CA WCC and through the CA EEM component.

A session lasts until you end it or the session's idle time exceeds the maximum timeout value for the master Tomcat server or the CA WCC component you are using (for example, the Quick Edit server or the Monitoring server). As a security measure, if the idle time exceeds the value specified in the Session Idle Time-Out property for the component in Configuration Manager, the session times out and you will be prompted to provide credentials to log back in to the component. A timeout can also be triggered if the CA WCC component is stopped and restarted.

CA WCC uses CA EEM to control access to functionality based on the role or roles with which a user is associated. This feature provides the ability to configure highly granular access to CA WCC and job administration features, functions, and commands.

Server Security

Workload automation server security requires that a user performing certain operations be authenticated. This can be accomplished through CA EEM for CA Workload Automation AE or with the user credential information provided in the CA WCC Credential application. If this requirement has not been satisfied, a login dialog may open when you attempt to initiate certain actions so that you can supply a server-authorized user ID and password.

Chapter 4: CA Workload Automation AE Native Security

This section contains the following topics:

Superusers

Under the native security model, users with administrative privileges are named as *superusers*. CA Workload Automation AE lets you define two levels of superusers—EDIT and EXEC. You can use the autosys_secure command to define these superusers.

Note: For more information about defining the superusers, see the *UNIX Implementation Guide or Windows Implementation Guide*.

EDIT Superuser

The EDIT superuser is the only user with permission to do the following:

- Edit or delete any job regardless of its owner or its permissions.
- Enter valid operating system user IDs and passwords in the database.

  These user IDs and passwords are required to log on to and run jobs on client computers. When an agent runs a job on a computer, it logs on as the user defined in the owner attribute for the job. To do this, the scheduler retrieves encrypted versions of the IDs and passwords for the *user@host_or_domain* and the *user@machine* from the event server and passes them to the agent.

  Note: Users who do not have EDIT superuser permission cannot add user IDs and passwords in the database. However, any user who knows an existing user ID and password can change that password or delete that user and password.

- Change the owner attribute of a job.

  The EDIT superuser can override user authentication (if enabled) on a job by job basis by changing the owner of the job from the *user@machine* form to the *user* form. When the owner of the job is specified in the *user* form, user authentication of the job at run time is not performed on the client computer.

  The purpose of the *user@machine* form is to prevent users from running jobs on machines where they do not have the appropriate permission. For example, root@machine prevents root on any machine from running jobs on all machines.

- Use the autosys_secure command to do the following:
  - Add or change Windows user IDs and passwords.
  - Change the database password.
  - Change the remote authentication method.

EXEC Superuser

The EXEC superuser is the only user with permission to do the following:

- Use the sendevent command or CA WCC to issue commands that affect the running or the state of a job.
- Enable external security.
- Use the sendevent command to stop the scheduler as follows:

  ```
  sendevent -E STOP_DEMON
  ```

Note: EXEC superuser privileges are typically granted to the night operator.

More Information:

Security on Events Sent by Users (see page 48)

Job Ownership

Every job runs under a specified user (the owner of the job). By default, the owner is the user who defines that job on a particular computer. The owner is defined by the owner job attribute and its value is specified as *user@machine*.

The default owner is automatically assigned when you define a job. However, if you have EDIT superuser privileges, you can override the default owner by specifying the owner attribute in the job definition.

UNIX:

The owner of the job is the user ID retrieved from the UNIX environment. The owner value is attached to the job definition in the form of *user@machine*. By default, only the owner can edit and execute the job.

The *user@machine* combination must have execute permission for any command specified in a job on the computer where the job is to run. The job owner must also have permission to access any device, resource, and so on, that the command must access. Therefore, the job owner must have the appropriate system permissions.

The owner's umask write permission is used as the default edit permission for the job, and the umask execute permission is used as the default execute permission for the job.

Windows:

The owner of the job (the Windows user ID that a job runs under and its corresponding password) must be defined in the database. The user ID and password are defined in the database by the EDIT superuser.

More information:

EDIT Superuser (see page 43)

User Types

To provide a level of security, CA Workload Automation AE associates the following four types of users with each job:

Owner

Creates the job.

Note: The owner of a job can let other users edit and execute the job by setting the permissions in the job definition.

Group

Resides in the same primary group as the owner.

Note: This is valid on UNIX only.

World

Specifies all users.

Machine

Specifies the users defined on a machine.

On UNIX, CA Workload Automation AE uses the user ID (uid) and group ID (gid) of the owner of a job to control the following:

- Who can edit, override, or delete a job definition.
- Who can execute the command specified in a job.

Permission Types

CA Workload Automation AE associates different types of permissions with each job. Every job has the following permission types:

Edit

Lets users edit, override, or delete a job definition.

Execute

Lets users use the sendevent command or CA WCC to send an execute event that affects the running or the state of a job.

Machine

Lets users who are logged on to a computer, other than the one where a job was created, edit or execute the job.

Note: For a job to run on a computer other than the one where it was defined, the owner of that job must have an account on that computer.

More information:

Security on Events Sent by Users (see page 48)

Granting Permissions

In the native security mode, individuals or groups of users are provided with edit and execute permissions on a job-by-job basis as follows:

- On UNIX, CA Workload Automation AE supports Owner, Group, and World Edit and Execute permissions.
- On Windows, CA Workload Automation AE supports Owner and World Edit and Execute permissions.

The owner of a job cannot override his or her ownership designation. By default, only the owner has edit and execute permissions for a job. All edit and execute permissions are valid only on the computer where the job is defined. Only the EDIT superuser has the permission to change the owner attribute for a job. However, the owner can use JIL to set the permission attribute in the job definition to grant other users edit and execute permissions for a job.

The following table shows the permissions that you can set using JIL:

JIL	Description
permission: gx	Valid on UNIX only. Grants execute permissions to all users in the job owner's primary group. Users in the group can execute the job if the user is logged on to the computer where the job was created (the computer specified in the owner attribute; that is, *user@machine*). **Note:** Group execute permissions are ignored in Windows job definitions. On Windows, the user executing the job must be the owner of the job, or world execute (wx) permissions must be specified for the job.
permission: ge	Valid on UNIX only. Grants edit permissions to all users in the job owner's primary group. Users in the group can edit the job if the user is logged on to the computer where the job was created (the computer specified in the owner attribute; that is, *user@machine*). **Note:** Group edit permissions are ignored in Windows job definitions. On Windows, the user editing the job must be the owner of the job, or world edit (we) permissions must be specified for the job.
permission: mx	Grants execute permissions to any authorized user, regardless of the computer they are logged on to. If this permission is not granted, the user must be logged on to the computer specified in the owner attribute (*user@machine*) to execute the job.

JIL	Description
permission: me	Grants edit permissions to any authorized user, regardless of the computer they are logged on to. If this permission is not granted, the user must be logged on to the computer specified in the owner attribute (*user@machine*) to edit the job.
permission: wx	Grants execute permissions to all users (world execute permissions). Any user can execute the job if the user is logged on to the computer where the job was created (the computer specified in the owner attribute; that is, *user@machine*).
permission: we	Grants edit permissions to all users (world edit permissions). Any user can edit the job if the user is logged on to the computer where the job was created (the computer specified in the owner attribute; that is, *user@machine*).

Note: A job and the command it runs always runs under the user specified in the owner attribute of the job definition. Execute permissions determine who can execute events against the job, but not the user that the job runs under. Even if world execute permissions (permission: we) are granted, the job still runs under the user defined in the owner attribute.

Security on Events Sent by Users

If you have the appropriate permissions, you can use the sendevent command to send the following execute events that affect the running of a job or the state of a job:

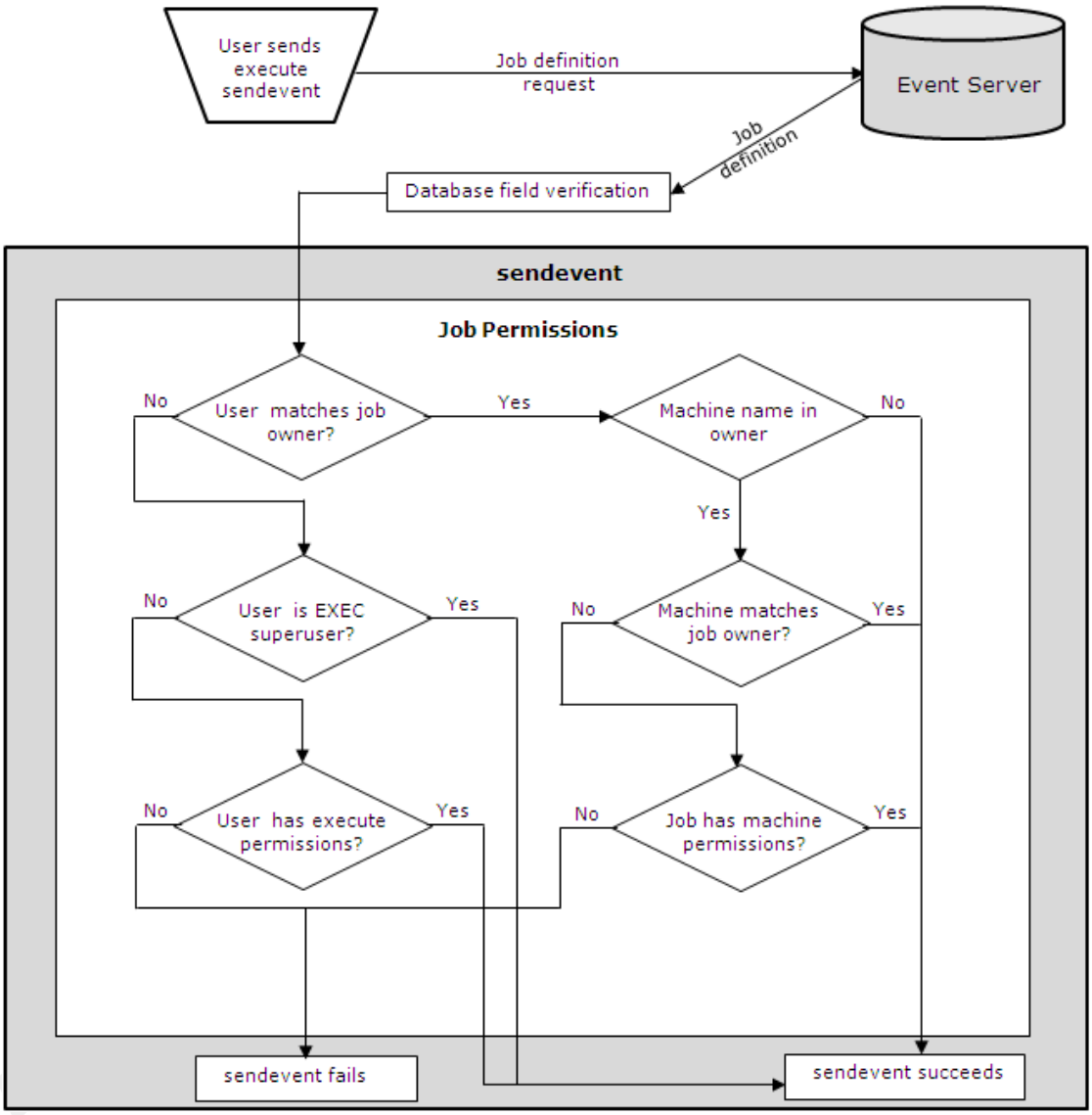
- CHANGE_PRIORITY
- CHANGE_STATUS
- DELETEJOB
- FORCE_STARTJOB
- JOB_OFF_HOLD
- JOB_OFF_ICE
- JOB_ON_HOLD
- JOB_ON_ICE
- KILLJOB
- SEND_SIGNAL
- STARTJOB

How Job Permissions are Verified

CA Workload Automation AE verifies the following when a user sends an event to start a job:

- Has the job definition been modified? If so, the job definition is invalid and the job does not run.
- Does the user match the owner as indicated in the job definition?
- Is the user the EXEC superuser as defined with autosys_secure?
- Does the user have job execute permissions as indicated in the job definition?
- Is there a machine name in the owner value of the job definition? The EDIT superuser can change the owner of the job from the *user@machine* form to the *user* form.
- Does the machine portion of the user log on credentials match the machine portion of the job owner definition?
- Does the job have machine permission as indicated by the job definition?

When you start a job by sending an event, the job permissions are verified as shown in the following illustration:



Chapter 5: CA Workload Automation AE External Security

This section contains the following topics:

How to Set Up External Security

This topic provides an overview of the steps that you must perform to set up external security by integrating CA Workload Automation AE with CA EEM.

Note: We recommend that you install CA EEM on a dedicated server and that you install CA EEM before you install CA Workload Automation AE. For more information about installing CA EEM, see the *CA Common Components Implementation Guide.*

To set up external security, follow these steps:

1. Register CA Workload Automation AE with CA EEM and create the default security policies. (see page 52)

   Note: You can skip this step if you selected the Create CA Embedded Entitlements Manager security policies for this instance check box during the CA Workload Automation AE installation.

2. Enable external security (see page 54).
3. Delegate administrative privileges on CA EEM to users who will add and remove policies (see page 55).
4. Create policies (see page 72).

   Note: For more information about policies and how to create them using the CA EEM web interface, see the *CA Embedded Entitlements Manager Programming Guide* and the *CA Embedded Entitlements Manager Online Help*.

Note: If you no longer use external security, you can disable it. (see page 82)

Register CA Workload Automation AE with CA EEM and Create Security Policies

CA EEM lets you centrally manage user access privileges and quickly deploy preconfigured basic security policies. During the CA Workload Automation AE installation, CA Workload Automation AE is registered with CA EEM and the security policies are created. If the security policies are not created during the installation or if you are installing multiple instances and want to create security policies for each of these instances, you can register CA Workload Automation AE with CA EEM and create the security policies using the as_safetool command.

Note: The preconfigured basic security policies grant all access modes to all users. So, we recommend that you use the CA EEM web interface to customize the default policies or create new policies and grant access modes based on your requirements.

To register CA Workload Automation AE with CA EEM and create security policies

1. Enter the following command at the UNIX operating system prompt or the Windows command prompt:

   ```
   as_safetool
   ```

   The following menu appears:

   ```
   Please select from the following options:
   [1] Manage eIAM Backend Server Location.
   [0] Exit CA WAAE IAM Toolkit Safetool Utility.
   ```

2. Enter 1 and press the Enter key.

   The following menu appears:

   ```
   Please select from the following options:
   [1] Set and Connect eIAM Backend Server Location as an Administrator.
   [2] Set eIAM Backend Server Location.
   [3] Show current Backend Server Location.
   [9] Exit from "Manage eIAM Backend Server Location" menu.
   [0] Exit CA WAAE IAM Toolkit Safetool Utility.
   ```

3. Enter 1 and press the Enter key.

4. Enter the CA EEM back end server host name, the application instance name, and the CA EEM administrator user name and password when prompted.

5. Enter 9 and press the Enter key after connecting to the CA EEM back-end server.

   The following menu appears:

   ```
   Please select from the following options:
   [1] Manage eIAM Backend Server Location.
   [2] Generate Instance Authentication Certificate.
   [3] Manage Instance Security Policy.
   [4] Manage Application Instance Administrators.
   [0] Exit CA WAAE IAM Toolkit Safetool Utility.
   ```

6. Enter 3 and press the Enter key.

   The following menu appears:

   ```
   Please select from the following options:
   [1] Install Default Security Policy.
   [2] Uninstall Default Security Policy.
   [3] Show Instances with Security Policy Installed.
   [4] Perform Asset Checks against Instance Security Policy.
   [9] Exit from "Manage Instance Security Policy" menu.
   [0] Exit CA WAAE IAM Toolkit Safetool Utility.
   ```

7. Enter 1 and press the Enter key.

8. Enter the CA Workload Automation AE instance name when prompted.

   The CA Workload Automation AE instance is registered with CA EEM and the security policies are created.

Enable External Security

Important! We recommend that you enable external security after installation by configuring CA Workload Automation AE to work with CA EEM. By default, CA Workload Automation AE uses native security. However, the level of protection that native security mode provides is limited compared to that of external security. Only external security lets you control role-based access to objects (such as jobs, calendars, cycles, global variables, machines, and resources) at a granular level.

An EXEC superuser in the native security mode can enable external security using the autosys_secure command and perform the basic CA EEM administration operations.

Note: After you enable external security, the job-level security and superuser privileges supported in the native security mode are no longer active. The policies in the as-control resource class govern who can disable external security.

To enable external security

1. Log on to CA Workload Automation AE as the EXEC superuser and enter the following command at the UNIX operating system prompt or the Windows command prompt:

   ```
   autosys_secure
   ```

   The following menu appears:

   ```
   Please select from the following options:
   [1] Activate EEM instance security.
   [2] Manage EDIT/EXEC superusers.
   [3] Change database password.
   [4] Change remote authentication method.
   [5] Manage user@host users.
   [6] Get Encrypted Password.
   [0] Exit CA WAAE Security Utility.
   ```

2. Enter 1 and press the Enter key.
3. Enter the CA EEM back end server host name when prompted.

   External security is enabled. The following message appears:

   ```
   CAUAJM_I_60201 EEM instance security successfully set.
   ```

Note: For more information about the autosys_secure command, see the *Reference Guide*.

Delegation of Administrative Privileges

During the CA Workload Automation AE installation, a repository is created in the CA EEM back-end server for the CA Workload Automation AE instance. This repository contains the policies that reside on the CA EEM server and visible to CA Workload Automation AE. Only users with access rights to the CA Workload Automation AE instance can connect to the repository through the CA EEM web interface and add or remove policies.

Only an authorized administrator can assign access rights to the CA Workload Automation AE application to a user. This can be done in *one* of the following ways:

- Add an CA EEM administrative scoping policy.
- Make the user a member of the CA Workload Automation AE Admin group.

Note: For more information about scoping policies and how to create them using the CA EEM web interface, see the *CA Embedded Entitlements Manager Programming Guide* and the *CA Embedded Entitlements Manager Online Help*.

CA Workload Automation AE Identities

During the CA Workload Automation AE installation, when CA Workload Automation AE registers with CA EEM, the WorkloadAutomationAE application instance is created on the CA EEM back-end server. This application instance stores the CA Workload Automation AE user details, access policies, calendars, and user groups and folders.

The WorkloadAutomationAE application instance includes the WorkloadAutomationAEAdmin user group. By default, this user group does not include any users; however, if granted access to it, you can add user groups to the WorkloadAutomationAEAdmin user group.

Note: The WorkloadAutomationAE application instance includes a default scoping policy. This scoping policy lets users added to the WorkloadAutomationAEAdmin user group modify policies defined in the resource classes.

You can define users or user groups at the global level in the Global application instance. Users created at the global level can be referenced by any application instance. At the global level, you can also configure CA EEM to reference all users from an external directory source, such as Active Directory. However, if you configure CA EEM to reference users from an external directory source, you cannot reference the global users previously created, nor can you create new users at the global level. User administration in that case is always managed at the external source.

You can manage your identities using the Manage Identities tab of CA EEM.

CA Workload Automation AE Resource Classes

A resource is a logical or physical entity whose access is controlled by CA EEM. You define groups of resources, called resource classes, to identify resources of similar types. A resource class contains the following information:

- Name
- Actions (List)
- Named Attributes (List)

Resource classes are used in the WorkloadAutomationAE application instance to classify the CA Workload Automation AE objects. You can create policies in each resource class to control user access to the corresponding CA Workload Automation AE object. For example, you can create policies for the as-job resource class to control access to jobs. For global variables, you can create policies in the as-gvar resource class.

Resource Class Details

The WorkloadAutomationAE application instance contains the following CA Workload Automation AE resource classes:

- as-appl
- as-calendar
- as-cycle
- as-control
- as-group
- as-gvar
- as-job
- as-joblog
- as-jobtype
- as-list
- as-machine
- as-owner
- as-resource

Access Modes

CA Workload Automation AE uses one or more of the following access modes on each of the resource classes:

- READ
- CREATE
- DELETE
- EXECUTE
- WRITE

The use of these access modes is explained in more detail in the description of each class.

as-appl Resource Class

The as-appl resource class specifies whether you can include the application attribute in a job definition and controls which jobs can be included in the job set.

The as-appl resource class has the following access modes:

READ

Controls whether you can view jobs that belong to a specific application or its contents. In READ mode, this resource class accepts the following command:

autorep

EXECUTE

Controls whether you can issue the sendevent command for the job that belongs to a specific application.

WRITE

Controls whether you can create or update jobs that belong to a specific existing application. In WRITE mode, this resource class accepts the following command:

jil

insert_job, update_job (using the application attribute)

as-calendar Resource Class

The as-calendar resource class controls access to calendars.

The as-calendar resource class has the following access modes:

READ

Controls whether you can view a calendar or its contents. In READ mode, this resource class accepts the following command:

autocal_asc

LIST CALENDAR DATES

Note: If as-list\AUTOCAL access is granted, you can view calendars.

CREATE

Controls whether you can create a calendar. In CREATE mode, this resource class accepts the following command:

autocal_asc

CREATE CALENDAR

DELETE

Controls whether you can delete a calendar. In DELETE mode, this resource class accepts the following command:

autocal_asc

DELETE CALENDAR

EXECUTE

Controls whether you can specify a calendar to run or to exclude in a job. In EXECUTE mode, this resource class accepts the following command:

jil

run_calendar, exclude_calendar

WRITE

Controls whether you can an update an existing calendar. In WRITE mode, this resource class accepts the following command:

autocal_asc

MODIFY CALENDAR

Note: Objects in this class can only contain the following characters: a-z, A-Z, 0-9, period (.), underscore (_), hyphen (-), and pound (#). Objects in this class cannot contain spaces.

More Information:

as-list Resource Class (see page 67)
Authorizing Users to View a List of Objects (see page 80)

as-control Resource Class

The as-control resource class controls access to critical CA Workload Automation AE services.

The as-control resource class has the following access mode:

EXECUTE

Controls whether you can issue the sendevent (-e STOP_DEMON) and the autosys_secure commands to control critical resources. In EXECUTE mode, the following default policies are created under the as-control resource class:

STOP_DEMON

Controls whether you can issue the sendevent command to stop the scheduler.

SECADM

Controls whether you can issue the autosys_secure command to disable external security.

EPLOG

Controls whether you can retrieve the scheduler log files from the application server.

SENDEVENT_GRPAPP

Controls whether you can issue the sendevent command for the group (-B) or application (-I).

as-cycle Resource Class

The as-cycle resource class controls access to cycles. A cycle is a list of one or more date ranges and is used in the definition of advanced rules for generating calendar dates.

Note: For more information about cycles, see the CA Workload Automation AE *Reference Guide*.

The as-cycle resource class has the following access modes:

READ

Controls whether you can view a cycle or its contents. In READ mode, this resource class accepts the following command:

autocal_asc

LIST CYCLE DATES

Note: If as-list\AUTOCAL access is granted, you can view cycles.

CREATE

Controls whether you can create a cycle. In CREATE mode, this resource class accepts the following command:

autocal_asc

CREATE CYCLE

DELETE

Controls whether you can delete a cycle. In DELETE mode, this resource class accepts the following command:

autocal_asc

DELETE CYCLE

WRITE

Controls whether you can update an existing cycle. In WRITE mode, this resource class accepts the following command:

autocal_asc

MODIFY CYCLES

Note: Objects in this class can only contain the following characters: a-z, A-Z, 0-9, period (.), underscore (_), hyphen (-), and pound (#). Objects in this class cannot contain spaces.

as-group Resource Class

The as-group resource class specifies whether you can include the group attribute in a job definition and controls which jobs can be included in the job set.

The as-group resource class has the following access modes:

READ

Controls whether you can view jobs that belong to a specific group or its contents. In READ mode, this resource class accepts the following command:

autorep

EXECUTE

Controls whether you can issue the sendevent command for the job that belongs to a specific group.

WRITE

Controls whether you can create or update jobs that belong to a specific existing group. In WRITE mode, this resource class accepts the following command:

jil

insert_job, update_job (using the group attribute)

as-gvar Resource Class

The as-gvar resource class controls access to global variables.

Note: Global variables are controlled using the sendevent command. So, the access modes are verified only during the execution of the sendevent command.

The as-gvar resource class has the following access modes:

READ

Controls whether you can view specific global variables. In READ mode, this resource class accepts the following commands:

autorep

-g variable

autostatus

-g variable

sendevent

CREATE

Controls whether you can create a global variable. In CREATE mode, this resource class accepts the following command:

sendevent

-g (new variable)

DELETE

Controls whether you can delete a global variable. In DELETE mode, this resource class accepts the following command:

sendevent

-g variable=DELETE

EXECUTE

Controls whether you can issue the sendevent command against all global variables simultaneously. In EXECUTE mode, this resource class accepts the following command:

sendevent

-e SET_GLOBAL, all-g options

WRITE

Controls whether you can update an existing global variable. In WRITE mode, this resource class accepts the following command:

sendevent

-g (existing variable)

as-job Resource Class

The as-job resource class controls access to jobs.

The as-job resource class has the following access modes:

READ

Controls whether you can view a job or its contents. In READ mode, this resource class accepts the following commands:

autorep

-J job, -q

Note: The as-list\AUTOREP controls whether you can issue the autorep command. If you do not have permissions, you cannot issue the autorep command. However, if you have the permission to issue the autorep command, but are not granted READ access mode, you cannot view jobs.

autostatad

-J job

Note: The as-list\AUTOSTAT controls whether you can issue the autostatad command. If you do not have permissions, you cannot issue the autostatad command. However, if you have the permission to issue the autostatad command, but are not granted READ access mode, you cannot view adapter jobs.

autostatus

-J job

job_depends

-J job

Note: The as-list\JOBDEP controls whether you can issue the job_depends command. If you do not have permissions, you cannot issue the job_depends command. However, if you have the permission to issue the job_depends command, but are not granted READ access mode, you cannot view the dependencies and conditions of jobs.

monbro

When as-list\MONBRO denied

Note: The as-list\MONBRO controls whether you can issue the monbro command. If you do not have permissions, you cannot issue the monbro command. However, if you have the permission to issue the monbro command, but are not granted READ access mode, you cannot view monitors or reports.

CREATE

Controls whether you can create a job. In CREATE mode, this resource class accepts the following command:

jil

insert_job

DELETE

Controls whether you can delete jobs directly or by issuing the sendevent command. In DELETE mode, this resource class accepts the following commands:

jil

delete_job

sendevent

-e DELETEJOB

EXECUTE

Controls whether you can issue the sendevent command for the job. In EXECUTE mode, this resource class accepts the following command:

sendevent

-e STARTJOB

-e KILLJOB

-e FORCE_STARTJOB

-e JOB_ON_ICE

-e JOB_OFF_ICE

-e JOB_ON_HOLD

-e JOB_OFF_HOLD

-e COMMENT -J job

WRITE

Controls whether you can update an existing job. In WRITE mode, this resource class accepts the following commands:

jil

update_job

sendevent

-e CHANGE_PRIORITY

More Information:

as-list Resource Class (see page 67)
Authorizing Users to View a List of Objects (see page 80)

as-joblog Resource Class

The as-joblog resource class controls access to job log files. These job log files include:

- Files that contain normal or error output from a job run (as defined by the std_out_file and std_err_file job attributes).
- Files that contain output from the job's command or the job's profile files.

The as-joblog resource class has the following access mode:

READ

Controls whether you can retrieve job log files from the application server.

Note: No spaces are allowed between the >> characters and the full path or file name in the std_out_file or std_err_file fields in a job definition.

as-jobtype Resource Class

The as-jobtype resource class controls access to job types.

The as-jobtype resource class has the following access modes:

READ

Controls whether you can view a job type or its contents. In READ mode, this resource class accepts the following commands:

autorep

-Y job_type

Note: The as-list\AUTOREP controls whether you can issue the autorep command. If you do not have permissions, you cannot issue the autorep command. However, if you have the permission to issue the autorep command, but are not granted READ access mode, you cannot view job types.

job_depends

-J job

Note: The as-list\JOBDEP controls whether you can issue the job_depends command. If you do not have permissions, you cannot issue the job_depends command. However, if you have the permission to issue the job_depends command, but are not granted READ access mode, you cannot view the dependencies and conditions of job types.

CREATE

Controls whether you can create a job type. In CREATE mode, this resource class accepts the following command:

jil

insert_job_type

DELETE

Controls whether you can delete job types directly. In DELETE mode, this resource class accepts the following command:

jil

delete_job_type

Note: When installed, the as_jobtype resource class is defined with the DELETE action. Its default policy is defined without the DELETE action. This prevents you from inadvertently deleting user-defined job types. You can update the default policy to include the DELETE action or you can define a new policy that grants access to delete user-defined job types.

EXECUTE

Controls whether you can specify a job type in a job. In EXECUTE mode, this resource class accepts the following command:

jil

job_type (with a value other than b, c, or f)

WRITE

Controls whether you can update an existing job type. In WRITE mode, this resource class accepts the following command:

jil

update_job_type

as-list Resource Class

The as-list resource class controls whether programs are directed to bypass individual security for read-only operation of a potentially large list of objects where the information displayed does not constitute a security violation (for example, in autorep).

Note: By using the default of this resource class, CA Workload Automation AE does not incur the overhead associated with issuing a security call for each individual line item displayed.

The as-list resource class has the following access mode:

READ

Controls whether programs are directed to bypass security. In READ mode, this resource class controls security pass using the following:

AUTOREP

Controls read access for the autorep command. This value is ignored for autorep reports created with the –q option. The command has the following security checkpoints:

-m ALL, -J ALL, -J box, -G ALL

AUTOSTAT

Controls read access for the autostatad command. The command has the following security checkpoint:

-J %

MONBRO

Controls read access for the monbro command. The command has the following security checkpoint:

Event related to secured jobs.

JOBDEP

Controls read access for the job_depends command. The command has the following security checkpoints:

-c –J ALL, -c –J %, -t %, -d %, -t ALL, -d ALL, -t box, -d box

as-machine Resource Class

The as-machine resource class controls access to machines, including whether you can use a machine object in a job definition.

The as-machine resource class has the following access modes:

READ

Controls whether you can view a machine or its contents. In READ mode, this resource class accepts the following command:

autorep

-m machine

Note: The as-list\AUTOREP controls whether you can issue the autorep command. If you do not have permissions, you cannot issue the autorep command. However, if you have the permission to issue the autorep command, but are not granted READ access mode, you cannot view machines.

CREATE

Controls whether you can create a machine definition. In CREATE mode, this resource class accepts the following command:

jil

insert_machine

DELETE

Controls whether you can delete a machine definition. In DELETE mode, this resource class accepts the following command:

jil

delete_machine

EXECUTE

Controls whether you can specify a machine in a job definition. In EXECUTE mode, this resource class accepts the following commands:

jil

machine

sendevent

-e STARTJOB

-e KILLJOB

-e FORCE_STARTJOB

-e JOB_ON_ICE

-e JOB_OFF_ICE

-e JOB_ON_HOLD

-e JOB_OFF_HOLD

-e COMMENT -J job

WRITE

Controls whether you can update a machine definition. In WRITE mode, this resource class accepts the following command:

jil

update_machine

as-owner Resource Class

The as-owner resource class specifies whether you can include the owner attribute in a job definition and controls which owners can be specified in the job definition. By default, the owner is the user who defines that job on a particular computer. However, when a user other than the one who defined the job is to be used as the owner, the as_owner resource class verifies whether that user can be specified in the job definition.

The as-owner resource class has the following access mode:

EXECUTE

Controls whether you can specify a different user as the owner of the job. In EXECUTE mode, this resource class accepts the following command:

jil

owner

as-resource Resource Class

The as-resource class controls access to resources.

The as-resource resource class has the following access modes:

READ

Controls whether you can view a resource or its contents. In READ mode, this resource class accepts the following commands:

autorep

job_depends

CREATE

Controls whether you can create resources. In CREATE mode, this resource class accepts the following command:

jil

insert_resource

DELETE

Controls whether you can delete resources. In DELETE mode, this resource class accepts the following command:

jil

delete_resource

EXECUTE

Controls whether you can specify a resource in a job definition. In EXECUTE mode, this resource class accepts the following commands:

jil

insert_resource, update_resource

WRITE

Controls whether you can update resources. In WRITE mode, this resource class accepts the following command:

jil

update_resource

CA Workload Automation AE Policies

During the CA Workload Automation AE installation, when CA Workload Automation AE is registered with CA EEM, the default security policies are created. These default security policies are included in the corresponding resource classes within the WorkloadAutomationAE application instance.

Note: The default security policies grant all access modes to all users. Therefore, we recommend that you use the CA EEM web interface to customize the default policies or create new policies and grant access modes based on your requirements.

Before performing an action on a specified object, CA Workload Automation AE issues a security call to the appropriate resource class in the repository. For example, for jobs, CA Workload Automation AE queries policies in the as-job resource class. For global variables, CA Workload Automation AE queries policies in the as-gvar resource class.

Policy Customization

CA EEM provides highly granular and flexible capabilities for creating customized policies to reflect the requirements of your enterprise. So, we recommend that you use the CA EEM web interface to customize the default policies or create new policies and grant access modes based on your requirements.

Create an Access Policy in CA EEM

You must create policies in CA EEM to control the user access to a CA Workload Automation AE object. CA EEM determines whether policies apply to the particular user by matching identities, resources, resource classes, and evaluating the filters.

Note: In CA EEM, you must create the policies under the appropriate resource class in the WorkloadAutomationAE application instance.

To create a policy in CA EEM

1. Open a browser and go to the following web site:

   http:*localhost*:5250\spin\eiam

   localhost

   Specifies the IP address or host name of the computer where CA EEM is installed.

   The CA EEM login page appears.

2. Select WorkloadAutomationAE instance from the Application drop-down list, enter EiamAdmin in the User Name field and the appropriate password in the Password field, and click Log In.

   The CA EEM web interface opens. The Home tab is displayed by default.

3. Select the Manage Access Policies tab.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name check box is selected by default.

4. Expand the Access Policies folder in the left pane.

   All the resource class folders are displayed.

5. Click the New Access Policy icon against the resource class folder where you want to create the policy.

   The New Access Policy dialog appears.

6. Complete the fields on this dialog as appropriate. Following are descriptions of fields that are not self-explanatory:

   Explicit Deny

   Specifies whether the policy explicitly denies the access that is specified in the policy. If you select this check box, the policy is listed under the Explicit Denies tab.

   Specifies whether the user is denied access explicitly by the policy.

   Disabled

   Specifies whether the policy is disabled.

   Pre-Deployment

   Specifies whether the policy is considered inactive. If you select this check box, the policy is not considered for permission check of identities.

   Assign Labels

   Specifies whether to assign a label to a policy.

   Calendar

   Specifies the calendar to be used during the match phase of policy evaluation. If no calendar is specified, all days and times match.

   Identities

   Specifies a list of identities (users, user groups, and global user groups) to be used during the match phase of policy evaluation. If this list is empty, all identities match.

   Resources

   Specifies a list of resources to use during the match phase of policy evaluation. If this list is empty, all resource names match.

   Actions

   Specifies the actions to use during the match phase of policy evaluation. You can grant read, write, create, delete, and execute access. If no actions are selected, all actions match.

   Filters

   Specifies the filters to use during the evaluate phase of policy evaluation.

7. Click Save.

   A policy is created.

Note: For more information about identities, resources, actions, filters, the match or evaluate phase of a policy, see the *CA Embedded Entitlements Manager Online Help*.

Resource Naming Convention

You can create policies in each resource class to control the user access to the corresponding CA Workload Automation AE object. The resource name in a policy consists of the CA Workload Automation AE instance name, a period, and the name of the corresponding object. For example, suppose that you want to update the payroll job in the ACE instance. CA Workload Automation AE queries CA EEM as follows:

```
as-job ACE.payroll
```

The as-owner resource class is an exception to this rule because it does not require the instance name. For example, suppose that you want to update the job owner field of a job to parrot@jungle in the ACE instance. CA Workload Automation AE queries CA EEM as follows:

```
as-owner parrot@jungle
```

Note: The security administrator must use the *instance*.*object* convention when creating policies except as cited for the as-owner class. You can use wildcards (for example,*) to create policies that apply to multiple objects across different CA Workload Automation AE instances. Also, CA EEM supports the use of regular expressions to define an object in a policy.

More Information:

Filtering and Regular Expressions (see page 16)

Authorizing a Request from a CA Workload Automation AE Client

Security policies are enforced by the application server, which obtains policy updates from the CA EEM back-end server.

Note: Because the scheduler and the agent do not enforce security, policy changes do not affect objects entered in the database. For example, if the security administrator withdraws a user's permission to create jobs, CA Workload Automation AE continues to run jobs created by the user before the change.

The time of authorization of a request from a CA Workload Automation AE client is important because the CA EEM security policies can have calendar dates and times associated with them. To properly authorize a request from a CA Workload Automation AE client, the application server must know what time is relative to the client.

The following occurs when a client makes a request to the application server:

- The client calculates its offset (in seconds) from the GMT time zone and sends it to the application server with the request.
- When the application server receives the request, it checks its offset (in seconds) from the GMT time zone.
- The application server subtracts the client's offset from its own to obtain the time zone difference (in seconds) from the client.
- The application server applies the difference to the current time and uses this as the time in its authorization check. This time represents the actual time relative to the client's time zone.

Authorizing Users to Create an Object

Only authorized users can create CA Workload Automation AE objects. To authorize users to create a CA Workload Automation AE object, you must create policies in CA EEM.

Note: The creation of certain CA Workload Automation AE objects, for example jobs, may require that the user is granted access to multiple resources.

Example: Authorize a User to Create a Job

Suppose that a user in your company wants to create a job. You must do the following:

- Authorize the user to create a job by creating a policy in the as_job resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific job name
Actions	create

- Authorize the user to specify the computer where the job runs as the machine attribute of the job by creating a policy in the as-machine resource class.You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific machine name
Actions	execute

- (Optional) Authorize the user to specify an application or group name in the application or group attribute of the job by creating a policy in the as-appl or as-group resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	application or group name
Actions	execute

- (Optional) Authorize the user to specify the run_calendar attribute of the job by creating a policy in the as-calendar resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific calendar name
Actions	execute

Note: This step is mandatory if the calendar name is also specified in the exclude_calendar attribute.

- (Optional) Authorize the user to specify another user in the owner attribute of the job by creating a policy in the as-owner resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	job owner **Note:** The job owner is not specific to a CA Workload Automation AE instance.
Actions	execute

More Information:

Create an Access Policy in CA EEM (see page 72)

Authorizing Users to Update an Object

Only authorized users can update CA Workload Automation AE objects. To authorize users to update a CA Workload Automation AE object, you must create policies in CA EEM.

Note: The update of certain CA Workload Automation AE objects, for example jobs, may require that the user is granted access to multiple resources.

Example: Authorize a User to Update a Job

Suppose that a user in your company wants to update a job. You must do the following:

- Authorize the user to update a job by creating a policy in the as_job resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific job name
Actions	write

- Authorize the user to specify the computer where the job runs as the machine attribute of the job by creating a policy in the as-machine resource class.You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific machine name
Actions	execute

- (Optional) Authorize the user to specify an application or group name in the application or group attribute of the job by creating a policy in the as-appl or as-group resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	application or group name
Actions	execute

- (Optional) Authorize the user to specify the run_calendar attribute of the job by creating a policy in the as-calendar resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific calendar name
Actions	execute

Note: This step is mandatory if the calendar name is also specified in the exclude_calendar attribute.

- (Optional) Authorize the user to specify another user in the owner attribute of the job by creating a policy in the as-owner resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	job owner **Note:** The job owner is not specific to a CA Workload Automation AE instance.
Actions	execute

Authorizing Users to Delete an Object

Only authorized users can delete CA Workload Automation AE objects. To authorize users to delete a CA Workload Automation AE object, you must create policies in CA EEM.

Example: Authorize a User to Delete a Job

Suppose that a user in your company wants to delete a job. You must authorize the user to delete a job by creating a policy in the as_job resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific job name
Actions	delete

Authorizing Users to View a List of Objects

When you issue the CA Workload Automation AE reporting commands, such as autocal_asc, autorep, or monbro, authorization checks are performed for every requested object before it is returned in the report. You can create policies in CA EEM to authorize users to bypass individual authorization checks and grant access to view every CA Workload Automation AE object in a listed output.

The as-list resource class lets authorized users bypass the individual authorization checks for each object when the autocal_asc, autorep, or monbro commands are issued, thus improving performance.

The list of objects is displayed based on the following conditions:

- If as-list READ access is granted, all objects are displayed with no further authorization checks.
- If as-list READ access is denied, only objects that are granted READ access are displayed. Objects that are denied READ access are not displayed. If every object in the list is denied READ access, nothing is displayed.
- For box jobs, an as-list READ access authorization check is also issued to view the contents of a box job as follows:
  - If as-list READ access is granted, information about the box job and all jobs in it are displayed.
  - If as-job READ access to the box job is denied, neither the box job nor the jobs in it are displayed.
  - If a box job is granted as-job READ access, but one or more jobs in the box are denied as-job READ access, only the box job and jobs in the box that are granted as-job READ access are displayed. Jobs in the box that are denied as-job READ access are not displayed.

Example: Authorize a User to View a Report of All Jobs Using the autorep Command

Suppose that a manager in your company wants to view a report of all jobs. You must authorize the user to bypass individual job authorization checks by creating a policy in the as_list resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific as-list resource name that is specific to the autorep command. For example, ACE.AUTOREP, where ACE is the CA Workload Automation AE instance name and AUTOREP is the pre-determined resource name qualifier specific to the autorep command.
Actions	read

Example: Authorize a User to View a Report of All Calendars Using the autocal_asc Command

Suppose that a manager in your company wants to view a report of all calendars. You must authorize the user to bypass individual calendar authorization checks by creating a policy in the as_list resource class. You must provide the following information:

CA EEM Field	Value
Identity	User
Resources	CA Workload Automation AE instance-specific as-list resource name that is specific to the autocal_asc command. For example, ACE.AUTOCAL, where ACE is the CA Workload Automation AE instance name and AUTOCAL is the pre-determined resource name qualifier specific to the autocal_asc command.
Actions	read

Disable External Security

If you want to run CA Workload Automation AE in native security mode, you must disable external security using the autosys_secure command.

Notes:

- Only the EDIT superuser can use the autosys_secure command to disable external security.
- You can control whether the EDIT superuser can disable external security by using the SECADM policy in the as_control resource class on CA EEM.

To disable external security

1. Log on to CA Workload Automation AE as the EDIT superuser and enter the following command at the UNIX operating system prompt or the Windows command prompt:

   ```
   autosys_secure
   ```

   The following menu appears:

   ```
   Please select from the following options:
   [1] Revert to NATIVE instance security.
   [2] Regenerate EEM certificate.
   [3] Change database password.
   [4] Change remote authentication method.
   [5] Manage user@host users.
   [6] Get Encrypted Password.
   [0] Exit CA WAAE Security Utility.
   ```

2. Enter 1 and press the Enter key.

   External security is disabled. The following message appears:

   ```
   CAUAJM_I_60202 NATIVE instance security successfully set.
   ```

3. (Optional) Remove the default security policies (see page 83).

   ```
   The default security policies are removed and the CA Workload Automation AE
   instance is unregistered from CA EEM.
   ```

Note: For more information about the autosys_secure command, see the *Reference Guide*.

Remove the Default Security Policies

CA Workload Automation AE can run either in external security mode through CA EEM or native security mode. By default, CA Workload Automation AE runs in native security mode.

During the CA Workload Automation AE installation, if you select the Create CA Embedded Entitlement Manager security policies for this instance check box, the default security policies are created. After the default security policies are created, you can enable external security to run CA Workload Automation AE in external security mode.

Later, if you do not want to run CA Workload Automation AE in external security mode and want to unregister the CA Workload Automation AE instance that is registered with CA EEM, you must disable external security and remove the default security policies.

To remove the default security policy

1. Enter the following command at the UNIX operating system prompt or the Windows command prompt:

   ```
   as_safetool
   ```

   The following menu appears:

   ```
   Please select from the following options:
   [1] Manage eIAM Backend Server Location.
   [0] Exit CA WAAE IAM Toolkit Safetool Utility.
   ```

2. Enter 1 and press the Enter key.

   The following menu appears:

   ```
   Please select from the following options:
   [1] Set and Connect eIAM Backend Server Location as an Administrator.
   [2] Set eIAM Backend Server Location.
   [3] Show current Backend Server Location.
   [9] Exit from "Manage eIAM Backend Server Location" menu.
   [0] Exit CA WAAE IAM Toolkit Safetool Utility.
   ```

3. Enter 1 and press the Enter key.

4. Enter the CA EEM back end server host name, the application instance name, and the CA EEM administrator user name and password when prompted.

5. Enter 9 and press the Enter key after connecting to the CA EEM back-end server.

   The following menu appears:

   ```
   Please select from the following options:
   [1] Manage eIAM Backend Server Location.
   [2] Generate Instance Authentication Certificate.
   [3] Manage Instance Security Policy.
   [4] Manage Application Instance Administrators.
   [0] Exit CA WAAE IAM Toolkit Safetool Utility.
   ```

6. Enter 3 and press the Enter key.

   The following menu appears:

   ```
   Please select from the following options:
   [1] Install Default Security Policy.
   [2] Uninstall Default Security Policy.
   [3] Show Instances with Security Policy Installed.
   [4] Perform Asset Checks against Instance Security Policy.
   [9] Exit from "Manage Instance Security Policy" menu.
   [0] Exit CA WAAE IAM Toolkit Safetool Utility.
   ```

7. Enter 2 and press the Enter key.

8. Enter the CA Workload Automation AE instance name when prompted.

   The CA Workload Automation AE instance is unregistered with CA EEM and the default security policy is removed.

Chapter 6: CA WCC Security

This section contains the following topics:

CA WCC Identities

CA WCC installs a number of default users. You can use the default users when you configure your CA WCC environment initially.

CA EEM can then be configured to reference additional users from an external directory source, such as Active Directory. However, if you configure CA EEM to reference users from an external directory source, the default users cannot log in to CA WCC unless you add them to that external directory source.

You can manage your identities using the Manage Identities tab of CA EEM.

Users

The following CA EEM users are created by default during the CA WCC installation:

EJMADMIN

Provides system configuration access for CA WCC. Typically, these users are system administrators or security administrators:

User name: ejmadmin

Password: ejmadmin

EJMEXEC

Provides access to summary information on job status for CA WCC. Typically, these users are executives and managers from other disciplines:

User name: ejmexec

Password: ejmexec

EJMOPERATOR

Provides a basic level of access for CA WCC. Typically, these users are console operators:

User name: ejmoperator

Password: ejmoperator

EJMSCHEDULER

Provides a level of access that includes the ability to create and modify jobs and job objects. Typically, these users are both schedulers and job and jobset administrators:

User name: ejmscheduler

Password: ejmscheduler

EJMSUPERVISOR

Provides a higher level of access. Typically, these users are scheduling or monitoring supervisors:

User name: ejmsupervisor

Password: ejmsupervisor

EJMCOMMANDER

Provides the highest level of access. These users have access to all in CA WCC features. Typically, these users are system administrators, scheduling monitors, or job and jobset administrators:

User name: ejmcommander

Password: ejmcommander

These users are set up in CA EEM. Each of these users has a default display in CA WCC, letting them access the applications to which they have rights.

Typically, you would use either the system administrator (ejmadmin) or the superuser (ejmcommander) to perform system configuration.

More information:

Create CA EEM Users (see page 129)
Configure CA EEM to Reference Active Directory Global Users (see page 91)

User Groups

The CA EEM installation creates default users using the policies defined in CA WCC. Although CA EEM can provide access authentication that is set up for individuals, the best practice for policy creation is to assign groups to policies and include individual users in those groups, instead of assigning individual users directly to the policies. This enables you to maintain stable policies even when individuals' access rights change.

You can assign any of the following types of groups:

Global Group

Includes users across all applications in the current CA EEM instance.

Application Group

Includes users in the currently selected application only.For example, you can create an application group based on a specific business function, such as Payroll. The users will be included in the group only if they are part of this function.

Dynamic Groups

Includes users based on their characteristics. For example, you can create a dynamic group based on parameters such as office, city, and title. The users will be automatically included in the group only if they meet the parameters set.

Note: For more information, see the CA Embedded Entitlements Manager documentation.

More Information:

Create CA EEM User Groups (see page 128)

User Roles

A number of preconfigured user roles provided with CA WCC are used in the default CA EEM policy definitions. These roles are defined as follows:

Console Operator

The *console operator* is responsible for monitoring job streams, correcting minor errors that are resolved through forcing job status or through one time overrides of data. The console operator also reports significant application errors to other operators and to the application owners. To complete these tasks, console operators use Job Status Console to monitor the jobs for which they are responsible and, optionally, may use Monitoring to track job status. They may also use the CPM (Critical Path Monitoring) application to ensure that job streams are running on time, the Event Console to view events related to the jobs for which they are responsible, and Quick View to view the detailed properties for a particular job in the flow.

Scheduler

The *scheduler* is responsible for the creation and maintenance of the job definitions. The schedulers use Quick Edit and Application Editor to create, modify, and delete jobs, calendars, and other job-related information such as constraints. Schedulers may also monitor job streams to make sure that they are running properly. Schedulers receive reports about problems with jobs that have already run, and they might need to look at historical information to analyze and correct a job stream.

Supervisor

The *supervisor* role performs all the tasks of a scheduler, and an additional level of tasks to help oversee and assist all the schedulers. The supervisor may also perform other tasks such as defining job streams, monitoring views, and job status views.

System Administrator

The *system administrator* is responsible for configuring and maintaining the CA WCC environment. This includes defining the scheduling manager servers and event servers, configuring software-specific properties, and defining host connection links for mainframe systems. System administrators may also be responsible for security, and for creating users and user groups assigned to specific roles.

Executive

The *executive*, or *manager*, has a very different set of needs compared to the other roles. The executive's tasks include monitoring, at the highest level, the overall status of the scheduling managers operating in their area. They generate and view reports, but do not build their own views. The executive uses the Job Status Views application to track the status of their views. The executive may also use the CPM (Critical Path Monitoring) application in conjunction with their job status views.

Security Administrator

The *security administrator* is responsible for configuring and maintaining security, and creating users and user groups assigned to specific roles. The security administrator also creates and maintains policies for CA EEM.

Commander

The *commander* is a superuser created as a convenience, and is a member of all the application groups.

The following table summarizes the functionalities that are available to the various user roles:

Role	Task	Data Item	Application Group	Default User
Console Operator	Monitor, report	Job Stream, View, Definition	Console Operator	ejmoperator
Supervisor	Define, monitor, report, react, correct	Job Stream, View, Definition, Criteria, Capacity, Monitors	Supervisor	ejmsupervisor
Scheduler	Define, analyze, update	Jobs, Schedules, JCL, Applications, Software, Constraints	Scheduler	ejmscheduler
System Administrator	Configure	Servers, Users	Administrator	ejmadmin
Executive	Plan, monitor, report, supervise	Users, Systems, Data-Center	Executive	ejmexec
Security Administrator	Security	Servers, Users, Objects	Security Administrator	ejmadmin and ejmcommander
Commander	Superuser	All	Commander	ejmcommander

Active Directory Authentication

An active directory is a directory structure used on Microsoft Windows-based systems to store information. This includes resources such as users and user groups. You can use CA EEM to configure access to the users already created in Active Directory.

Notes:

- Default CA WCC users cannot log in to CA WCC unless you add them to the external directory source.
- The default port for Active Directory is 389. For improved performance, you can use port 3268.

Configure CA EEM to Reference Active Directory Global Users

Configuring CA EEM to reference the Active Directory global users enables the applications using CA EEM, such as CA Workload Automation AE and CA WCC, to share the user information from the Active Directory.

Note: You may want to configure filters so that users do not access all entries in the global Active Directory.

To configure CA EEM to reference the Active Directory global users

1. Access the CA EEM Home page.
2. Click the Configure tab.

   The Configure page opens.
3. Click the EEM Server link.

   The EEM Server page appears.
4. Click Global Users/Global Groups.

   The EEM Server Configuration page appears in the right pane.
5. Select the Reference from an external directory option.

   Additional fields relating to the external directory appear.
6. Leave the default directory server type as Microsoft Active Directory, and complete the following fields:
   a. Enter the host computer on which Active Directory is running.
   b. Enter the port for eiamAdmin. The default is 389.

      Note: Enter the port only if the Active Directory Server administrator has reconfigured the default Active Directory Server port.
   c. Enter the Base DN. The value specified here must be similar to the base DN value that is specified in the Active Directory server.
   d. Enter the User DN that is used to attach to the eiamAdmin server.
   e. Enter the required value in the Password field.
   f. Re-enter the value in the Confirm Password field.
   g. Enter the values in the remaining fields based on the enterprise configuration.
7. Click Save.

   The values you have specified are saved, and CA EEM is configured to the Active Directory.

More information:

Log in to CA EEM for CA WCC (see page 128)

Retrieve Users that Belong to a Particular Group

By default in CA EEM, all the users defined in the configuration are eligible for authentication and retrieval. To limit the users due to the size of the directory or to restrict unnecessary users from gaining access, you can apply a filter.

LDAP filters use the following syntax rules:

- The segments of the expressions must be enclosed in parentheses.
- The following relational operators can be applied: <. <=, =, >=, and >.
- Expressions are formed using the prefix operators & and |.

To retrieve users belonging to a particular group

1. Click the Configure tab.

   The Configure page opens.
2. Select Custom Mapped Directory from the Type drop-down list.
3. Edit the User Filter field as follows:
   a. Enter (& at the beginning of the existing filter.
   b. Enter) at the end of the 'objectClass=computer))' string.
   c. Enter (memberOf=groupName)).

      For example, to include users belonging to the JAWS group, enter the following:

      ```
      (&(&(objectClass=user)(!(objectClass=computer)))(memberOf=CN=JAWS,OU=Groups,OU=EMEA,DC=ca,DC=com))
      ```

      Note: The 'memberOf' string must contain the fully qualified name for the group.
4. Click Save.

   The changes are saved and the status icon turns to green.
5. Select the Manage Identities tab and then click the Users link.

   The Search Users section is displayed.
6. Search for users and verify that you can view the users as defined in the filter.

   The users that belong to the specified group are retrieved.

Retrieve Groups Using a Particular Pattern in Group Names

By default in CA EEM, all the groups and folders are cached. If you have a large number of groups present in the LDAP directory, then CA EEM caches the folders and groups. This, in turn, causes iTechnology memory to increase, resulting in a longer time requirement for CA EEM to load the data. To alleviate the problem, you can retrieve groups using a pattern in their group names.

To retrieve groups using a particular pattern in the group name

1. Click the Configure tab.

   The Configure page opens.

2. Select Custom Mapped Directory from the Type drop-down list.

3. Set Group Attribute to Description and Directory Attribute to description in the Global Group Attributes section.

4. Select User Group as Attribute from the Group Type drop-down list

   The Group Membership Attribute and the User Membership Attribute values default to memberOf.

5. Edit the Group Filter field to include the groups to be included.

   For example, to include a single group pattern, the string may resemble the following:

   ```
   (&(objectClass=group)(sAMAccountName=T*))
   ```

   To include multiple group patterns, the string may resemble the following:

   ```
   (&(objectClass=group)(sAMAccountName=T*)(sAMAccountName=Acct*)(sAMAccountName
   =Pay*)))
   ```

6. Click Save.

   The changes are saved and the status icon turns to green.

7. Select the Manage Identities tab and then click the Groups link.

   The Search Groups section is displayed.

8. Search for groups and verify that you can view the groups as defined in the filter.

   The groups that match the pattern entered in the Group Filter field are retrieved.

Retrieve Groups Without Caching

By default in CA EEM, all the groups and folders are cached. If you have a large number of groups in the LDAP directory, then you can retrieve groups without caching.

To retrieve groups without caching

1. Click the Configure tab.

   The Configure page opens.

2. Select Custom Mapped Directory from the Type drop-down list.
3. Select the Include Unmapped Attributes check box.
4. Set Group Attribute to Description and Directory Attribute to description in the Global Group Attributes section.
5. Select Use Group as Container from the Group Type field, and then enter an incorrect attribute, such as testing, in the Unique Member Attribute field.
6. Set objectClass= to an invalid value, such as ggggg, in the Group Filter field.
7. Click Save.

   The changes are saved and the status icon turns green.

   Select the Manage Identities tab.

8. Click the Groups link, accept the default values, and click Go.

   No groups are listed in the User Groups section.

9. Click Users, accept the default values, and click Go.

   The users are listed in the Users section.

10. Select a user from the list.

    The User attributes are displayed in the right pane.

11. Click  .

    The Unmapped Attribute Details section is displayed.

12. Locate the memberOf attribute under the Attribute Name section and copy the value from the Attribute Value field.
13. Select the Manage Access Policies tab..

    The page opens, displaying all of the existing policies in the left pane.

14. Click  to create a new Dynamic User Group policy.

    The new policy displays in the right pane.

15. Enter a name for the new policy in the Name field.
16. Select Access Policy from the Type field.
17. Click Add Filter in the Access Policy Configuration section.

18. Select global user from the Left type drop-down list and then select … from the value drop-down list
19. Select value from the Right type drop-down list.
20. Paste the string you copied in Step 14 in the value field and click Save.

    The new policy is saved.
21. Select the Manage Identities tab.
22. Click the Users link, accept the default values, and click Go.

    The users are listed in the Users section.
23. Select the same user from the list.

    The user attributes are displayed in the right pane.
24. Expand the Extended User Group Membership section in the right pane.

    The Dynamic User Groups assigned to that user are displayed.

CA WCC Resource Classes

A resource is a logical or physical entity whose access is controlled by CA EEM. You define groups of resources, called resource classes, to identify resources of similar types. A resource class contains the following information:

- Name
- Actions (List)
- Named Attributes (List)

Eleven resource classes are available for the CA WCC application. The classes can be categorized into the following types:

Access Resource Classes

The ApplicationAccess and ServerAccess classes control access to applications and servers respectively. The ConfigurationControl class lets you configure additional levels of access to certain application features, such as those in Credentials and High Availability.

Action or Log Resource Classes

The JobActionAutoSys, AlertAction, LogAction, CommandSetup, and CommandExecute classes control access to job actions, alert actions, job and scheduler logs, and Enterprise Command Line commands respectively.

Object Access Resource Classes

The MonitorViewControl, ObjectAccess, and ObjectControl classes are used to control specific types of access to a variety of objects that are displayed and maintained by CA WCC. Each of the actions that can be performed on these objects is enumerated in the resource actions. The name of the object is specified for the resource name. There are typically attributes specified for server type and server name. These resource classes are only implemented in server-side code.

Application Access

The ApplicationAccess resource controls access to the applications that operate in the CA WCC framework. Most of these applications are available as tabs from the main page of CA WCC; however, Job Status Console, Event Console, and the Tutorials are opened in a separate browser window when you click the appropriate link in the top right corner of the main page, and the Job Status Views and Command Line Console applications are displayed on the Main tab. If the user does not have access to an application, that application or the link to that application is not displayed. Each application is represented by an action in the resource.

Notes:

- In order to view and/or edit job properties from Monitoring, Job Status Console, and Resources, you must also have application access to Quick View and Quick Edit respectively.
- To edit job properties from Quick View, you must also have application access to Quick Edit.

Resource Name Value

Specifies an application category and, if there are or can be multiple applications in that category, may additionally indicate a particular application. The resource name value is structured as follows:

category/*name*

- *category*—the category for a set of applications
- *name*—a specific application within the application category

Examples:

HighAvailability

Configuration/Security

Reports/Forecast

Actions

The following actions are available:

Action Code	Default Resource Name Mapping
Framework	Framework
Configuration	Configuration
SecurityConfiguration	Configuration/Security
HighAvailability	HighAvailability
Credentials	Credentials
Tutorials	Tutorials

Action Code	Default Resource Name Mapping
JobStatusConsole	JobStatus/Console
EventConsole	EventConsole
QuickView	QuickView
CommandLine	CommandLine
QuickEdit	QuickEdit
ApplicationEditor	ApplicationEditor
Resources	Resources
Monitoring	Monitoring
ReportsForecast	Reports/Forecast
CPM	CPM
JobStatusViews	JobStatus/Views
CommandLineConsole	CommandLine/Console

Attributes

component—the name of the component issuing the authorization check

category—the category for a set of applications

name—a specific application within the application category

More information:

Create an ApplicationAccess Policy (see page 130)
Application Security Table (see page 119)

Server Access

The ServerAccess resource identifies the servers that a user can access, including scheduling manager, event console, and JAWS servers. Each CA WCC component will only allow access to servers, if applicable, that the current user is permitted to access.

Note: A view can be edited only by a user who has access to all servers defined in the view.

Resource Name Value

Corresponds to the name specified for the server in CA WCC Configuration Manager. The resource name value is structured as follows:

server/*serverName*

- *serverName*—the name of the server as defined in CA WCC Configuration Manager

Default: server/*

Examples:

server/myAEserver

server/ECserver

Actions

The following action is available:

- Access

Attributes

component—the name of the component issuing the authorization check

type—CA Workload Automation AE, Event Console, or JAWS

serverName—the name of the server as defined in CA WCC Configuration Manager

More information:

Create a ServerAccess Policy (see page 132)
Application Security Table (see page 119)

Job Actions for CA Workload Automation AE

The JobActionAutoSys resource identifies the actions that can be performed on a CA Workload Automation AE job. It determines what should be displayed in the Actions drop-down list for a CA Workload Automation AE job.

Resource Name Value

Specifies a view or a server. The resource name value is structured as one of the following:

view/*viewName*

server/*serverName*

- *viewName*—the name of the view as defined in Monitoring or Job Status Console
- *serverName*—the name of the server as defined in CA WCC Configuration Manager

Defaults:

The following resources are used in Monitoring and Job Status Console:

- view/*
- server/*

The following resource is used in Quick Edit, Quick View, Event Console, and Resources:

- server/*

Examples:

view/myView

server/AEserver

Actions

The following actions are available for Monitoring, Quick View, and Job Status Console:

- CancelEvent
- ForceStart
- Kill
- OffHold
- OffIce
- OnHold
- OnIce
- SendEvent
- Start

- ReleaseResources
- Comment
- SendSignal
- ChangePriority
- ChangeStatus
- SetGlobal
- Reply

The following actions are available for Quick Edit:

- CreateOneTimeOverride
- DeleteOneTimeOverride

The following action is available for Resources:

- ReleaseResources

The following actions are available for Event Console:

- CancelEvent
- SendEvent

Attributes

component—the name of the component issuing the authorization check

nameType—View or Server

name—*viewName* or *serverName*

- viewName—the name of the view as defined in Monitoring or Job Status Console
- serverName—the name of the server as defined in CA WCC Configuration Manager

More information:

Create a JobActionAutoSys Policy (see page 143)
Application Security Table (see page 119)

Alert or Alarm Actions

The AlertAction resource identifies the actions that can be performed on an alarm or an alert in Job Status Console. Alarms and alerts are considered equivalent with respect to actions.

Resource Name Value

Specifies a view or a server. The resource name value is structured as one of the following:

view/*viewName*

server/*serverName*

- *viewName*—the name of the view as defined in Job Status Console
- *serverName*—the name of the server as defined in CA WCC Configuration Manager

Defaults:

view/*

server/*

Examples:

server/autosys1

view/myView

Actions

The following actions are available for Job Status Console:

- Acknowledge
- Close
- Open

Attributes

component—the name of the component issuing the authorization check

nameType—View or Server

name—*viewName* or *serverName*

- viewName—the name of the view as defined in Job Status Console
- serverName—the name of the server as defined in CA WCC Configuration Manager

More information:

Create an AlertAction Policy (see page 142)
Application Security Table (see page 119)

Log Access

The LogAccess resource identifies the logs that are accessible, by server, in Job Status Console, Monitoring, and Quick View.

Resource Name Value

Specifies a server. The resource name value is structured as follows:

server/*serverName*

- *serverName*—the name of the server as defined in CA WCC Configuration Manager

Default: server/*

Examples:

server/serverAE

server/SrvrACE

Actions

The following actions are available for Job Status Console:

- JobLog
- SchedulerLog

The following action is available for Monitoring:

- SchedulerLog

The following action is available for Quick View:

- JobLog

Attributes

component—the name of the component issuing the authorization check

serverName—the name of the server as defined in CA WCC Configuration Manager

More information:

Create a LogAccess Policy (see page 140)
Application Security Table (see page 119)

Command Setup

The CommandSetup resource identifies the actions that can be performed on an entry in the specified command lists of the Enterprise Command Line. It controls what the Enterprise Command Line allows during the specification, saving, and listing of commands.

This resource is used together with the CommandExecute resource to control what a user can do with the Enterprise Command Line. The CommandExecute resource makes a call to CA EEM for each command being executed. In addition, the Command Line back-end agent checks that command against the list of commands allowed to be run.

Resource Name Value

Specifies the command to use. The resource name value is structured as follows:

mode/*cmdName*

- *mode*—the mode to use:
  - Local
  - Global
- *cmdName*—the name of the command specification in the local or global command list

Defaults:

local/*

global/*

Examples:

local/ListMyJobs

global/CancelJob

Actions

The following actions are available:

- Create
- Modify
- Delete
- View

Attributes

component—the name of the component issuing the authorization check

mode—local or global

cmdName—the name of the command specification in the local or global command list

Notes:

- local/* limits users to defining and viewing commands in their own My Commands list.
- global/* + Action of View lets users view and execute commands in the global list, but not modify them.

More information:

Create a CommandSetup Policy (see page 145)
Application Security Table (see page 119)

Command Execution

The CommandExecute resource identifies the command to be executed on a given back-end server. The Enterprise Command Line View performs an authorization check prior to executing the requested command. A policy can be written to control who can perform a specified command on a specified server.

Resource Name Value

Specifies the command to execute. The resource name value is structured as follows:

server/*serverName*/*command*

- *serverName*—the name of the server as defined in CA WCC Configuration Manager
- *command*—command to be executed on the given server

Default: server/*

Example: server/AEserver/cmd-spec

Actions

The following action is available:

- Execute

Attributes

component—the name of the component issuing the authorization check

serverName—the name of the server as defined in CA WCC Configuration Manager

command—command to be executed on the given server

Note: The command value is the actual command with all the parameters. It might be somewhat difficult to parse the command and its parameters using CA EEM expression processing; however, it is usually easy to isolate the first word of the command, so it is possible to identify the name of the command being executed.

More information:

Create a CommandExecute Policy (see page 147)
Application Security Table (see page 119)

Monitor View Control

The MonitorViewControl resource controls the actions that can be performed on a particular monitoring view in Job Status Console and Monitoring.

Resource Name Value

Specifies the view to control. The resource name value is structured as follows:

view/*viewName*/JobFlow

- *viewName*—the name of the view as defined in CA WCC Monitoring or Job Status Console

Default: */JobFlow

Examples:

view/myview/JobFlow

view/FailedJobs/JobFlow

Actions

The following actions are available for Monitoring and Job Status Console:

- Create
- Modify
- Delete
- View

Attributes

component—the name of the component issuing the authorization check

viewName—the name of the view as defined in Monitoring or Job Status Console

More information:

Create a MonitorViewControl Policy (see page 138)
Application Security Table (see page 119)

Object Access

The ObjectAccess resource controls access to a particular object. Policies written against this class can control which objects a user can view in Reports-Forecast and Job Status Console.

Important! By default, ObjectAccess filtering is disabled. It must be enabled in CA WCC Configuration individually by server. ObjectAccess filtering can induce significant overhead if a server contains a large number of objects (for example, 50,000 job definitions or more). For this reason, you should manually activate ObjectAccess filtering for only those servers that you deem appropriate.

Resource Name Value

Specifies the server and object to control. The resource name value is structured as follows:

server/*serverName*/*objectType*/*objectName*

- *serverName*—the name of the server as defined in CA WCC Configuration Manager
- *objectType*—the type of object to control:
  - AlertPolicy
  - JobStatus
  - Forecast
- *objectName*—the name of the object

Defaults:

The following resources are used for Job Status Console:

- */AlertPolicy/*
- */JobStatus/*

The following resource is used for Reports-Forecast:

- */Forecast/*

Examples:

server/AEserver/AlertPolicy/FailedJobs

server/SrvrACE/Forecast/myReport

Actions

The following action is available for Reports-Forecast and Job Status Console:

- Access

Attributes

component—the name of the component issuing the authorization check

serverName—the name of the server as defined in CA WCC Configuration Manager

objectType—AlertPolicy, Forecast, or Job Status

objectName—the name of the object

More information:

Create an ObjectAccess Policy (see page 134)
Application Security Table (see page 119)

Object Control

The ObjectControl resource controls the actions that can be performed on a particular class of objects. Individual objects are not controlled. For example, a user must be authorized to create a job. The ObjectControl policy is used by Application Editor, Job Status Console, Quick Edit, and Reports-Forecast.

Resource Name Value

Specifies the server and the class of objects to control. The resource name value is structured as one of the following:

server/*serverName*/*objectType*

server/*serverName*/Job/*jobType*

server/*serverName*/Forecast/*reportName*

- *serverName*—the name of the server as defined in CA WCC Configuration
- *objectType*—the type of object to control:
  - AlertPolicy
  - Calendar
  - Cycle
  - ExtendedCalendar
  - GlobalVariable
  - Job
  - Forecast
- *jobType*—the job type for a Job object:
  - Box
  - Command
  - File Watcher
  - File Trigger
  - FTP
  - i5/OS
  - z/OS Regular
  - z/OS Data Set Trigger
  - z/OS Manual
  - CPU Monitoring
  - Disk Monitoring
  - IP Monitoring

  - Process Monitoring
  - Text File Reading and Monitoring
  - Windows Event Log Monitoring
  - Windows Service Monitoring
  - Oracle E-Business Suite Copy Single Request
  - Oracle E-Business Suite Request Set
  - PeopleSoft
  - SAP Batch Input Session
  - SAP BW InfoPackage
  - SAP Data Archiving
  - SAP Event Monitor
  - SAP Job Copy
  - SAP Process Monitor
  - SAP R/3
  - Database Monitor
  - Database Stored Procedure
  - Database Trigger
  - SQL
  - Entity Bean
  - HTTP
  - JMS Publish
  - JMS Subscribe
  - JMX-MBean Attribute Get
  - JMX-MBean Attribute Set
  - JMX-MBean Create Instance
  - JMX-MBean Remove Instance
  - JMX-MBean Subscribe
  - POJO
  - RMI
  - Session Bean
  - Web Service
- *reportName*—the name of the Forecast report

Defaults:

The following resources are used for Quick Edit:

- */GlobalVariable
- */Job*
- */Calendar
- */ExtendedCalendar
- */Cycle

The following resources are used for Application Editor:

- */GlobalVariable
- */Job*

The following resource is used for Job Status Console:

- */AlertPolicy

The following resource is used for Reports-Forecast:

- */Forecast/*

Examples:

server/AEserver/GlobalVariable

server/AEserver/Calendar

server/AEserver/Job/Command

server/AEserver/Job/JMX-MBeanSubscribe

server/AEserver/Job/UserDefined2

server/SrvrACE/Forecast/Week1

Actions

The following actions are available for Quick Edit, Application Editor, Job Status Console, and Reports-Forecast:

- Create
- Delete
- Modify

Attributes

component—the name of the component issuing the authorization check

serverName—the name of the server as defined in CA WCC Configuration Manager

reportName—the name of the Forecast report

objectType—AlertPolicy, Calendar, Cycle, ExtendedCalendar, GlobalVariable, Forecast, or Job

jobType—Box. Command, File Watcher, File Trigger. FTP, i5/OS, z/OS Regular, z/OS Data Set Trigger, z/OS Manual, CPU Monitoring, Disk Monitoring, IP Monitoring, Process Monitoring, Text File Reading and Monitoring, Windows Event Log Monitoring, Windows Service Monitoring, Oracle E-Business Suite Copy Single Request, Oracle E-Business Suite Request Set, PeopleSoft, SAP Batch Input Session, SAP BW InfoPackage, SAP Data Archiving, SAP Event Monitor, SAP Job Copy, SAP Process Monitor, SAP R/3, Database Monitor, Database Stored Procedure, Database Trigger, SQL, Entity Bean, HTTP, JMS Publish, JMS Subscribe, JMX-MBean Attribute Get, JMX-MBean Attribute Set, JMX-MBean Create Instance, JMX-MBean Remove Instance, JMX-MBean Subscribe, POJO, RMI, Session Bean, Web Service

More information:

Create an ObjectControl Policy (see page 136)
Application Security Table (see page 119)

Configuration Control

The ConfigurationControl resource controls the configuration of various applications within CA WCC, including Credentials, CPM (Critical Path Monitoring), High Availability, Framework, and Monitoring. This security resource class is different than other resource classes because it addresses a variety of disparate resources. This is done to minimize the number of classes that must be defined to protect various CA WCC resources. For that reason, the description of the class is separated into sections based on usage.

Resource Name Value

Specifies a particular application to be configured. The resource name value is structured as follows:

name/options

- *name*—one of the applications that makes use of this resource class; currently, this is "Credentials", "CPM", "HighAvailability", "Framework", or "Monitoring" to match the names specified for these applications in the action codes for the ApplicationAccess resource class
- *options*—application-specific configuration options

Actions

The actions available vary depending on the specified application. For more information, see the actions list for the appropriate application.

Attributes

component—the name of the component issuing the authorization check

name—the ApplicationAccess action code for a specific application

options—everything after the name section in the resource name, which is unique for each application utilizing the ConfigurationControl resource class

Credentials

For the Credentials application, this class controls the setting of credentials, as follows:

- Use credential - An authorization request is made for each use of a credential.
- Edit credential - An authorization request is made for each create/modify/delete of a credential.

Resource Name Value

Specifies the Credentials application and the options to configure for it. The resource name value is structured as follows:

Credentials/server/*serverName*/*userid*/*credentialtype*/*creduserid*

- *serverName*—the server name in a Credentials policy definition
- *userid*—the user ID of the user requesting the credential use or modification
- *credentialtype*—the credential type in a Credentials policy definition; currently, only "password" is supported but additional types are planned for future releases
- *creduserid*—the credential user ID in a Credentials policy definition

Default: Credentials/server/*

Examples:

Credential/server/ServerACE/brifr/password/brifr

Credential/server/AEserver/admin/password/opuser

Actions

The actions available vary depending on the specified option. The following actions are available:

- Create—specified when a credential is created; this lets you control who can create a credential
- Save—specified when a credential has been created or modified and a new value is being saved for it; this lets you control who can save a credential
- Delete—specified when a credential is being deleted; this lets you control who can delete a credential

CPM (Critical Path Monitoring)

For the CPM application, this class controls who can save personal and global filters and set the active filter.

Resource Name Value

Specifies the CPM application and the options to configure for it. The resource name value is structured as follows:

CPM/*object*

- *object*—indicates the object being configured, as follows:
  - MyFilter
  - GlobalFilter
  - ActiveFilter

Default: CPM/*object*

Examples:

CPM/MyFilter

CPM/ActiveFilter

Actions

The actions available vary depending on the specified option. The following actions are available:

- Save—specified when a personal filter (MyFilter) or global filter (GlobalFilter) has been created or modified and a new value is being saved for it; this lets you control who can save a personal or global filter, for example, in case you want to limit the ability to save global filters to a specific group of users but allow anyone to save a personal filter
- Set—specified when the active filter is being set; this lets you control who can activate a filter

High Availability

For the High Availability application, this class controls who can publish and apply high availability configurations.

Resource Name Value

Specifies the High Availability application and the options to configure for it. The resource name value is structured as follows:

HighAvailability/server/*serverName*

- *serverName*—the name of the core or spectator member of the High Availability group

Default: HighAvailability/*server*/*

Examples:

HighAvailability/server/Core1

HighAvailability/server/Spec2

Actions

The actions available vary depending on the specified option. The following actions are available:

- Apply—specified when a high availability configuration is applied to a server; this lets you control who can apply a configuration; the server specified is the spectator server to which the configuration is being saved
- Publish—specified when a high availability configuration is published to a server; this lets you control who can publish a configuration; the server specified is the spectator server to which the configuration is being published

Framework

For the Framework application, this class controls who can access specific themes.

Resource Name Value

Specifies the Framework application and the option to configure for it. The resource name value is structured as follows:

Framework/theme/*instanceName*/*themeName*

- *themeName*—the name of the theme
- *instanceName*—the name of the CA WCC server

Default: Framework/theme/*

Examples:

Framework/theme/Default

Framework/theme/AEServer/Theme2

Actions

The following action is available:

- Access

Monitoring

For the Monitoring application, this class controls Monitoring settings, as follows:

- Delete cache - controls who can drop all jobs in the database cache and rebuild the cache for the specified server. This action is accessed from Monitoring Batch Interface only.
- Use flow layout - Checks if the user has permission to access the flow layouts
- View job dependencies - controls who can view job dependencies for a specified view

Resource Name Value

Specifies the Monitoring application and the options to configure for it. The resource name value is structured as follows:

Monitoring/server/*serverName*/cache

Monitoring/view/*viewName*/JobDeps

- *serverName*—the name of the server as defined in CA WCC Configuration Manager
- *viewName*—the name of the view as defined in CA WCC Monitoring

Defaults:

Monitoring/server/.*/cache

Monitoring/view/.*/JobDeps

Monitoring/flow/layout

Examples:

Monitoring/server/SrvrACE/cache

Monitoring/server/AEserver/cache

Monitoring/view/MyView/JobDeps

Actions

The actions available vary depending on the specified option. The following actions are available:

- Delete—specified in the Monitoring Batch Interface, this lets you control who can delete the cache for a specified server.
- Access—specified when a job dependency is accessed, this lets you control who can view the job dependencies; specified when the flow layout is accessed, this lets you control who has access to change the layout from the default value.

More information:

Create a ConfigurationControl Policy (see page 149)
Application Security Table (see page 119)

CA WCC Policies

Policies define the access rights of a particular user or user group to a particular resource. They associate identities with resources. There is a single default policy for each resource class. The default policy for each class is delivered with the CA WCC application in CA EEM.

Application Security Table

By default, all user roles are granted access to the framework required to support CA WCC, the applications available from the Main page (either as tabs or from links), and the user tutorials.

The ApplicationAccess policy controls access to each CA WCC application, including the Framework, Main page, and tutorials. Each tab on the Main page can be hidden or displayed based on the authorizations you set up in this policy.

The ServerAccess policy controls access to the back-end server instances. Using this policy, you can restrict user roles to specific instances of the scheduling manager. By default, members of the Administrator, Commander, ConsoleOperator, Executive, Scheduler, SecurityAdministrator, and Supervisor roles have access to all configured servers.

The following table lists all available CA WCC applications and the CA EEM policies associated with each one:

CA WCC Application	CA EEM Policy	Usage
Application Editor Used to manage job flows and dependencies using a graphical, drag-and-drop user interface.	ApplicationAccess	Lets you limit access to only those people who will require access to Application Editor. By default, members of the Commander, Scheduler, and Supervisor roles are granted access to Application Editor.
	ObjectControl	Specifies the server and class of objects to control. Individual objects are not controlled. You can specify jobType (Command, J2EE) or objectType (global variable). By default, members of the Commander, Supervisor, and Scheduler roles have create, modify, and delete control for all Application Editor object classes.
Configuration Manager Used to configure CA WCC and to perform administrative tasks	ApplicationAccess	Lets you limit access to only those people who will be responsible for configuring your system. By default, members of the Administrator and Commander roles are granted access to

CA WCC Application	CA EEM Policy	Usage
related to job management and scheduling.		Configuration Manager.
Credentials Used to let users add, modify, and delete their credentials for back-end servers in real time.	ApplicationAccess	Lets you limit access to users who will be responsible for setting credentials for back-end servers. By default, members of the Administrator, Commander, ConsoleOperator, Executive, Scheduler, and Supervisor roles are granted access to Credentials.
	ConfigurationControl - Credentials (Admin)	Lets you set rules to which credentials must conform. By default, members of the Commander or Administrator roles are granted access to create, delete, and save credentials.
	ConfigurationControl - Credentials (User)	Lets you set access to allow users to update their own credentials. By default, members of the Administrator, Commander, ConsoleOperator, Executive, Scheduler, and Supervisor roles are granted access to create, delete, and save their personal credentials.
ECL (Enterprise Command Line) Used to run JIL commands from the Windows command line or UNIX console.	ApplicationAccess	Lets you limit access to only those people who will require access to the Enterprise Command Line. By default, members of the Administrator, Commander, ConsoleOperator, Scheduler, and Supervisor roles are granted access to the Enterprise Command Line.
	CommandExecute	Lets you restrict the commands that can be executed and who can perform those commands on a specific server. Using this policy, you could grant a user group access only to canceling jobs on the server. By default, members of the Administrator, Commander, ConsoleOperator, Scheduler, and Supervisor roles have execute access to all configured servers.
	CommandSetup	Specifies whether the user can view, create, modify, or delete commands based on command names. Using this policy, you can limit users to defining and viewing commands in their own My Commands list, or let them

CA WCC Application	CA EEM Policy	Usage
		view and execute commands in the global list, but not modify them. By default, members of the Commander and Supervisor roles have access to create, modify, delete, and view commands; members of the Administrator, ConsoleOperator, and Scheduler roles have view access only.
Event Console Used to view and respond to events sent by CA Workload Automation AE to the event agent.	ApplicationAccess	Lets you limit access to only those people who will require access to Event Console. By default, members of the Commander, ConsoleOperator, and Supervisor roles are granted access to Event Console.
	JobActionAutoSys	Specifies the actions that can be issued for a job. Using this policy, you could assign one user group the rights to send and cancel events for jobs, while assigning another the rights to send an event to a job only. You can restrict the rights further by specifying a particular server. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles have access to all actions for all servers.
Framework	ApplicationAccess	Defines the access to the framework of CA WCC, including the login page and content that appears above and below the tabbed pages. By default, everyone has access to Framework.
	ConfigurationControl - FrameworkThemes	Specifies the themes the user can access on the specified CA WCC instance. By default, members of the Administrator, Commander, ConsoleOperator, Executive, Scheduler, SecurityAdministrator, and Supervisor roles can access all themes.
High Availability Used to synchronize and monitor configurations between the members of a group of servers.	ApplicationAccess	Lets you limit access to only those people who will be responsible for synchronizing and monitoring configurations between the members of a group of servers. By default, members of the Administrator and Commander roles are granted access to High Availability.
	ConfigurationControl - HighAvailability	Controls access to High Availability configuration. Using this policy, you could assign apply and publish access to a core server and apply access only to a spectator

CA WCC Application	CA EEM Policy	Usage
		server. By default, members of the Commander and Administrator roles are granted deploy and apply access.
Job Status Console Used to monitor jobs and job alerts, and produce a summary of job status by server or custom view.	ApplicationAccess	Lets you limit access to only those people who will be responsible for creating and monitoring jobs and alerts. **Note:** You must have access to Quick View or Quick Edit to view the job details or to edit a job's properties. By default, members of the Commander, ConsoleOperator, Executive, Scheduler, and Supervisor roles are granted access to Job Status Console.
	AlertAction	Specifies the actions that can be performed on an alarm or alert. Using this policy, you could grant rights to acknowledge and close alerts for a specified view and server. By default, members of the Commander, ConsoleOperator, and Supervisor roles have acknowledge, close, and open access to all views and configured servers.
	JobActionAutoSys	Specifies the actions that can be issued for a job. Using this policy, you could assign one user group the rights to start and terminate jobs, while assigning another the rights to start a job only. You can restrict the rights further by specifying a particular server or view. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles have access to all actions for all servers and views.
	LogAccess	Specifies the logs that are accessible, by server. Using this policy, you could grant access to the job run log (JobLog) and the scheduler log (SchedulerLog) on a particular server. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles are granted access to the EventProcessorLog and the JobLog on all servers.
	MonitorViewControl	Controls the actions that can be performed on a job status view and the views to which users

CA WCC Application	CA EEM Policy	Usage
		have access. Using this policy, you could grant specific users create, modify, delete, and view access to all views with names that begin with a certain naming convention, such as Payroll. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles are granted create, modify, delete, and view, actions for all job status views.
	ObjectAccess	Specifies the object types the user can access. Using this policy, you could give a user access to specific alert policies by name. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles can access all alert policies and views.
	ObjectControl	Controls create, delete, or modify access to a class of objects. Individual objects are not controlled. You can specify jobType (Command, J2EE) or objectType (global variable, calendar). By default, members of the Commander, Supervisor, and Scheduler roles have create, modify, and delete control for all objects.
Monitoring Used to create, manage, and monitor job flow views.	ApplicationAccess	Lets you limit access to only those people who will be responsible for monitoring jobs and job flows. **Note:** You must have access to Quick View or Quick Edit to view the job details or to edit a job's properties. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles are granted access to Monitoring.
	JobActionAutoSys	Secures CA Workload Automation AE actions based on view name and server name. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles have access to all actions for all servers.
	LogAccess	Specifies the logs that are accessible, by server. Using this policy, you could grant access to the job run log (JobLog) and the scheduler log (SchedulerLog) on a particular server. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor

CA WCC Application	CA EEM Policy	Usage
		roles are granted access to the SchedulerLog and the JobLog on all servers.
	MonitorViewControl	Controls the actions that can be performed on a view and the views to which users have access. Using this policy, you could grant specific users create, modify, delete, and view access to all views with names that begin with a certain naming convention, such as Payroll. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles are granted create, modify, delete, and view, actions for all monitoring views.
	ConfigurationControl - Monitoring (Admin)	Controls user permission to delete the monitoring cache for a server. By default, members of the Commander and Administrator roles are granted access.
	ConfigurationControl - Monitoring (User)	Controls whether a user has access to change the layouts in the Flow section in Monitoring. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor are granted access.
Quick Edit Used to create and manage all of your workload objects, including jobs and calendars.	ApplicationAccess	Lets you limit access to only those people who will be responsible for creating, modifying, or deleting workload objects. By default, members of the Commander, Scheduler, and Supervisor roles are granted access to Quick Edit.
	JobActionAutoSys	Secures CA Workload Automation AE actions based on server name. If the override actions are not allowed, the icons and buttons to create or delete overrides are not rendered. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles have access to all actions for all servers.
	ObjectControl	Controls create, delete, or modify access to a class of objects. Individual objects are not controlled. You can specify jobType (Command, J2EE) or objectType (global variable, calendar). By default, members of the Commander, Supervisor, and Scheduler roles have create, modify, and delete control for all objects.

CA WCC Application	CA EEM Policy	Usage
Quick View Used to access consolidated job detail information for CA Workload Automation AE servers.	ApplicationAccess	Lets you limit access to only those people who will be responsible for monitoring jobs. **Note:** You must have access to Quick Edit to edit a job's properties. By default, members of the Commander, ConsoleOperator, Executive, Scheduler, and Supervisor roles are granted access to Quick View.
	JobActionAutoSys	Specifies the actions that can be issued for a job based on server name. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles have access to all actions for all servers.
	LogAccess	Specifies the logs that are accessible, by server. Using this policy, you could grant access to the job run log (JobLog) on a particular server. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles are granted access to the JobLog on all servers.
Reports-Forecast Used to forecast job completion on a CA Workload Automation AE server and to generate reports.	ApplicationAccess	Lets you limit access to only those people who will require access to Reports-Forecast. By default, members of the Commander, ConsoleOperator, Executive, Scheduler, and Supervisor roles are granted access to Forecast.
	ObjectAccess	Lets you limit access to only those people who will require access to Forecast reports. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles are granted access to Forecast reports.
	ObjectControl	Specifies whether the user can create, modify, or delete Forecast reports. By default, members of the Commander, Supervisor, and Scheduler roles have create, modify, and delete control for all Forecast reports.

CA WCC Application	CA EEM Policy	Usage
Resources Used to create and manage CA Workload Automation AE virtual resources, run resource reports, and view job resource usage.	ApplicationAccess	Lets you limit access to only those people who will be responsible for monitoring resources. **Note:** You must have access to Quick View or Quick Edit to view the job details or to edit a job's properties. By default, members of the Commander, Scheduler, and Supervisor roles are granted access to Resources.
	JobActionAutoSys	Specifies the actions that can be issued for a job based on server name. By default, members of the Commander, ConsoleOperator, Scheduler, and Supervisor roles have access to all actions for all servers.

More information:

Alert or Alarm Actions (see page 102)
Job Actions for CA Workload Automation AE (see page 100)
Log Access (see page 103)
Monitor View Control (see page 106)
Command Setup (see page 104)
Command Execution (see page 105)
Configuration Control (see page 113)
Server Access (see page 99)
Object Control (see page 109)
Object Access (see page 107)
Application Access (see page 97)

How to Customize Your CA WCC Policies

CA WCC installs a default policy for each resource class in the CA EEM application for CA WCC. The rights granted by these policies are designed to reflect typical user roles.

CA EEM provides highly granular and flexible capabilities for creating customized policies to reflect the requirements of your enterprise. The information in this section describes the default policy for each resource class and provides sample procedures to create an additional policy for the resource class, enabling you to customize your enterprise security. This section also includes information on completing permission checks. We recommend you review these procedures before proceeding with any customizations, so that you have all the materials and information you need to help ensure a successful CA EEM security setup.

Before making use of the sample policy procedures, you should perform the following steps:

1. Create views in Job Status Console and Monitoring, both with the same name.

   Note: For more information about creating views, see the *Job Status Console Help* and the *Monitoring Help*.

2. Log in to CA EEM using the CA WCC application.
3. Create a CA EEM Application group. The sample policies will use this Application Group.
4. Create one or more CA EEM users that are members of the application group created in Step 3.
5. Create the necessary policies, and perform a permission check after each.
6. Verify, in CA WCC, that your policy setup works properly.

Log in to CA EEM for CA WCC

You must be logged in to the CA EEM component of CA WCC before you complete any of the remaining steps of the policy creation process. To create a policy, you need to access CA EEM.

To access CA EEM from within CA WCC

1. Log in to CA WCC as a user who has security administrator rights.

   CA WCC opens to the user's default view.

2. Click the EEM link.

   A CA EEM login screen is displayed with EiamAdmin displayed in the User Name field.

3. Select the CA WCC application from the Application drop-down list, enter the appropriate password in the Password field, and click Log In.

   CA EEM opens with the Home tab displayed by default.

Create CA EEM User Groups

To give multiple users the same access rights, you can create user groups. We recommend you create multiple user groups by user role. For example, you could create one user group for schedulers and another for administrators. This procedure describes creating application groups; however, you can also create global groups, dynamic groups, and individual users.

To create CA EEM application groups

1. Click Manage Identities on the Home tab of the CA EEM home page.

   The Users and Groups links appear, with the Users link selected by default.

2. (Optional) Click Groups, leave the Show application groups check box selected, and click Go.

   All the available application groups are listed under Application Groups in the User Groups section.

3. Click the New Application Group icon in the left pane.

   The New Application User Group page appears in the right pane.

4. Enter a name for the new application group, and click Save.

   A confirmation message appears.

5. Repeat Steps 3 and 4 to create additional application groups as necessary.

Create CA EEM Users

To give users access to CA WCC, you must first create CA EEM users. All CA EEM users are global users. This procedure describes how to add CA EEM users manually, but you can also add users by referencing an Active Directory.

You can add users to global groups from any application. To add a user to an application group or a dynamic group, you must be logged into the specific CA EEM application that contains the application group or dynamic group. This procedure shows you how to add a user to an application group.

To create CA EEM users

1. Select the Manage Identities tab in the CA EEM application.

   The Users and Groups links appear, with the Users link selected by default.

2. Select the Application User Details option button.

3. (Optional) Leave User Name selected in the Attribute drop-down list, leave LIKE selected in the Operation drop-down list, leave the Value field blank, and click Go to view all the users currently available.

   All application users are listed under Users in the Users pane.

4. Click the New User icon in the left pane.

   The New User page appears in the right pane.

5. Enter a name for the new user in the Name field.

6. Click Add Application User Details.

7. Select the appropriate application groups from the Available User Groups box in the Application Group Membership section, and click  .

   The application groups are added to the Selected User Groups box.

8. Enter the password for the user in the New Password and Confirm Password fields in the Authentication section, and click Save.

   A confirmation message appears.

9. Repeat Steps 4 through 8 to create additional users as necessary.

Create an ApplicationAccess Policy

To enable a user to access specified applications reflecting that user's role, you can create an ApplicationAccess policy. For this policy, a type of Identity Access Control List is recommended. This example is written based on that selection.

The default policy, ApplicationAccessDefault, establishes different application access levels for each CA EEM group (role) installed with the CA WCC application. For example, only the Administrator and Commander (superuser) groups have access to the Configuration Manager application; whereas, the Commander, Console Operator, Scheduler, and Supervisor groups have access to the Monitoring application.

To create an ApplicationAccess policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click ApplicationAccess in the left pane.

   All the ApplicationAccess policies are listed in the right pane.

3. Click the New Access Policy icon  next to ApplicationAccess in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, and leave ApplicationAccess selected in the Resource Class Name drop-down list.

5. Select the Identity Access Control List option button in the Type field.

   A confirmation prompt appears.

6. Click OK.

   The Identity Access Control List Configuration section appears.

7. Select Application Group from the Type drop-down list in the Identity Access Control List Configuration section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

8. Click Search.

   A list of application groups is displayed.

9. Select the application group you created, and click  .

   The application group appears in the Selected Identities list.

10. Select the following check boxes under Actions for your group:
    - Framework
    - Monitoring
    - JobStatusConsole
    - QuickView
    - Credentials

    Important! Every user must have access to the Framework application.
11. Click Save at the top or bottom right of the page.

    Your new policy appears in the list.
12. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create a ServerAccess Policy

To enable a user to access specified servers reflecting that user's role, you can create a ServerAccess policy. You define each of the servers as a resource in the server/*serverName* format.

Note: You can create policies to reflect the naming convention that you use when you define the server name for your servers in CA WCC Configuration Manager. For example, you can create a policy that gives a user access to only the servers with the identifier AE*, as shown in the sample policy below.

For this policy, a type of Access Policy is recommended. This example is written based on that selection.

The default policy, ServerAccessDefault, lets all users access all servers.

To create a ServerAccess policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click ServerAccess in the left pane.

   All the ServerAccess policies are listed in the right pane.

3. Click the New Access Policy icon next to ServerAccess in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, leave ServerAccess in the Resource Class Name drop-down list, and leave the Access Policy option button selected in the Type field.

5. Select Application Group from the Type drop-down list in the Identities section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

6. Click Search.

   A list of application groups is displayed.

7. Select the application group you created, and click .

   The application group appears in the Selected Identities box to the right.

8. Enter server/AEServer in the Add Resource field, and click ⊕.

   The resource is added to the resource list, providing access to the AEServer server.

9. Enter server/AESrvr* in the Add Resource field, and click ⊕.

   Note: Creating a policy with this entry gives you access to all servers that start with the characters AESrvr. If you use a naming convention for your servers (for example, AESrvrACE, AESrvrAC2, AESrvrAC3), you can use a wildcard in the resource.

10. Select the Access check box under Actions in the Access Policy Configuration section for your group.

11. Click Save at the top or bottom right of the page.

    The policy is created.

12. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create an ObjectAccess Policy

To assign a user-specified view access to jobs and job objects, you can create an ObjectAccess policy. For this policy type, either the Access Policy or Access Control List type is recommended. If you want to permit the same access to all the resource types assigned in the policy, use the Access Policy type. This example is written based on that selection. If you want to permit different types of access to the resource types assigned in the policy, use the Access Control List type instead.

The default policy, ObjectAccessDefault, lets all users access all objects.

Note: To use an ObjectAccess policy for your Forecast and Job Status Console objects, the Filter Object setting on the server properties page in Configuration Manager must be enabled for the appropriate CA Workload Automation AE server. This setting instructs Forecast and Job Status Console whether to check for CA EEM policies when retrieving objects so, if it is not enabled, ObjectAccess policies do not apply.

To create an ObjectAccess policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click ObjectAccess in the left pane.

   All the ObjectAccess policies are listed in the right pane.

3. Click New Access Policy icon  next to ObjectAccess in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, leave ObjectAccess in the Resource Class Name drop-down list, and leave the Access Policy option button selected in the Type field.

5. Select Application Group from the Type drop-down list in the Identities section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

6. Click Search.

   A list of application groups is displayed.

7. Select the application group you created, and click  .

   The application group appears in the Selected Identities box to the right.

8. Enter */AlertPolicy/* in the Add Resource field in the Access Policy Configuration section, and click .

   The resource is added to the resource list.

9. Enter */JobStatus/* in the Add Resource field, and click  .

   The resource is added to the resource list.

   Note: To provide view access to a subset of the objects available in each of these categories, you can use the final asterisk (*) as a wildcard, and insert an identifier. For example, to view only the alert policies whose names begin with ACC* (the ACC application alert policies), enter */AlertPolicy/ACC*.

10. Select the Access check box under Actions for your group.
11. Click Save at the top or bottom right of the page.

    The policy is created.

12. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create an ObjectControl Policy

To assign a user specified levels of access to the various objects available, you can create an ObjectControl policy. For this policy, a type of either Access Policy or Access Control List is recommended. If you want to permit the same access to all the resource types assigned in the policy, use the Access Policy type. This example is written based on that selection. If you want to permit different types of access to the resource types assigned in the policy, use the Access Control List type instead.

The default policy, ObjectControlDefault, lets users in the Scheduler, Supervisor, and Commander (superuser) groups control all objects.

To create an ObjectControl policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click ObjectControl in the left pane.

   All the ObjectControl policies are listed in the right pane.

3. Click New Access Policy icon next to ObjectControl in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, leave ObjectControl in the Resource Class Name drop-down list, and leave the Access Policy option button selected in the Type field.

5. Select Application Group from the Type drop-down list in the Identities section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

6. Click Search.

   A list of application groups is displayed.

7. Select the application group you created, and click .

   The application group appears in the Selected Identities box to the right.

8. Enter */Forecast/* in the Add Resource field in the Access Policy Configuration section, and click .

   The resource is added to the resource list.

9. Enter */Job* in the Add Resource field, and click ⊕.

   The resource is added to the resource list.

10. Select the Create and Modify check boxes under Actions for your group.
11. Click Save at the top or bottom right of the page.

    The policy is created.

12. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create a MonitorViewControl Policy

To enable a user to access specified job flows, you can create a MonitorViewControl policy. For this policy, a type of Identity Access Control List is recommended. This example is written based on that selection. It assumes that you have a Job Status Console view and a Monitoring view with the same name and that the user should have view-only access to both (no create, modify, or delete rights).

The default policy, MonitorViewControlDefault, lets users in the Console Operator, Scheduler, Supervisor, and Commander (superuser) groups control all objects.

To create a MonitorViewControl policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear across the top, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click MonitorViewControl in the left pane.

   All the MonitorViewControl policies are listed in the right pane.

3. Click the New Access Policy icon next to MonitorViewControl in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, and leave MonitorViewControl in the Resource Class Name drop-down list.

5. Select the Identity Access Control List option button in the Type field.

   A confirmation prompt appears.

6. Click OK.

   The Identity Access Control List Configuration section appears.

7. Select Application Group from the Type drop-down list in the Identity Access Control List Configuration section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

8. Click Search.

   A list of application groups is displayed.

9. Select the application group you created, and click  .

   The application group appears in the Selected Identities list.

10. Enter view/*viewName*/JobFlow (where *viewName* indicates the name of the applicable job flow) in the Add Resource field in the Resources section, and click  .

    The resource is added to the resource list, providing access to that job flow.

11. Select the View check box under Actions for your group.
12. Click Save at the top or bottom right of the page.

    The policy is created.

13. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create a LogAccess Policy

To enable a user to access specified logs, you can create a LogAccess policy. For this policy, a type of Identity Access Control List is recommended. This example is written based on that selection.

The default policy, LogAccessDefault, lets users in the Console Operator, Scheduler, Supervisor, and Commander (superuser) groups view all logs.

To create a LogAccess policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies left selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click LogAccess in the left pane.

   All the LogAccess policies are listed in the right pane.

3. Click the New Access Policy icon next to LogAccess in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, and leave LogAccess selected in the Resource Class Name drop-down list.

5. Select the Identity Access Control List Type option button in the Type field.

   A confirmation prompt appears.

6. Click OK.

   The Identity Access Control List Type Configuration section appears.

7. Select Application Group from the Type drop-down list in the Identity Access Control List Configuration section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

8. Click Search.

   A list of application groups is displayed.

9. Select the application group you created, and click .

   The application group appears in the Selected Identities list.

10. Enter server/AEServer in the Add Resource field in the Resources section, and click .

    The resource is added to the resource list, providing access to the AEServer server.

11. Select the JobLog check box under Actions for your group.

12. Click Save at the top or bottom right of the page.

    The policy is created.

13. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create an AlertAction Policy

To enable a user to access specified alert actions, you can create an AlertAction policy. For this policy, a type of Access Policy is recommended. This example is written based on that selection. It assumes that you have a Job Status Console view, and describes giving access to acknowledge and close alerts for that view and the AEServer server.

The default policy, AlertActionDefault, lets users in the Console Operator, Supervisor, and Commander (superuser) groups acknowledge, close, and re-open alerts and alarms.

To create an AlertAction policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click AlertAction in the left pane.

   All the AlertAction policies are listed in the right pane.

3. Click the New Access Policy icon next to AlertAction in the left pane.

   The New Access Policy tab appears in the right pane.

4. Enter a unique policy name in the General pane, leave AlertAction selected in the Resource Class Name drop-down list, and leave the Access Policy option button selected in the Type field.

5. Select Application Group from the Type drop-down list in the Identities section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

6. Click Search.

   A list of application groups is displayed.

7. Select the application group you created, and click .

   The application group appears in the Selected Identities box to the right.

8. Enter view/*viewName* (where *viewName* indicates the name of the applicable Job Status Console view) in the Add Resource field in the Access Policy Configuration section, and click .

   The resource is added to the resource list, providing access to performing actions on alerts/alarms in that Job Status Console view.

9. Enter server/AEServer in the Add Resource field, and click .

   The resource is added to the resource list, providing access to performing actions on alerts/alarms from the AEServer server.

10. Select the Acknowledge and Close check boxes under Actions for your group.

11. Click Save at the top or bottom right of the page.

    The policy is created.

12. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create a JobActionAutoSys Policy

To enable a user to access specified CA Workload Automation AE job actions, you can create a JobActionAutoSys policy. For this policy, a type of Identity Access Control List is recommended. This example is written based on that selection.

The default policy, JobActionAutoSysDefault, lets users in the Console Operator, Supervisor, Scheduler, and Commander (superuser) groups perform the selected actions.

To create a JobActionAutoSys policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click JobActionAutoSys in the left pane.

   All the JobActionAutoSys policies are listed in the right pane.

3. Click the New Access Policy icon  next to JobActionAutoSys in the left pane.

   The New Access Policy tab appears in the right pane.

4. Enter a unique policy name in the Name field, and leave JobActionAutoSys selected in the Resource Class Name drop-down list.

5. Select the Identity Access Control List Type option button in the Type field.

   A confirmation prompt appears.

6. Click OK.

   The Identity Access Control List Type Configuration section appears.

7. Select Application Group from the Type drop-down list in the Identity Access Control List Type Configuration section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

8. Click Search.

   A list of application groups is displayed.

9. Select the application group you created, and click  .

   The application group appears in the Selected Identities list.

10. Enter view/*viewName* (where *viewName* indicates the name of the applicable Job Status Console view) in the Add Resource field in the Resources section, and click  .

    The resource is added to the resource list, providing access to performing actions on jobs in that Job Status Console view.

11. Enter server/AEServer in the Add Resource field, and click  .

    The resource is added to the resource list, providing access to performing actions on jobs from the AEServer server that are included in job flows.

12. Select the Start, OnHold, and OffHold check boxes under Actions for your group.

13. Click Save at the top or bottom right of the page.

    The policy is created.

14. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create a CommandSetup Policy

To enable a user to create and save commands in the Enterprise Command Line application (or Command Line Console), you can create a CommandSetup policy. For this policy, a type of Identity Access Control List is recommended. This example is written based on that selection and assumes that the users in the group will have rights to view, create, and modify commands in all tabs, but will not be able to delete commands.

The default policy, CommandSetupDefault, lets users in the Supervisor and Commander (superuser) groups view, create, modify, and delete commands, and lets users in the Administrator, Console Operator, and Scheduler groups view commands but not create or delete them.

To create a CommandSetup policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click CommandSetup in the left pane.

   All the CommandSetup policies are listed in the right pane.

3. Click the New Access Policy icon next to CommandSetup in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, and leave CommandSetup in the Resource Class Name drop-down list.

5. Select the Identity Access Control List option button in the Type field.

   A confirmation prompt appears.

6. Click OK.

   The Identity Access Control List Configuration section appears.

7. Select Application Group from the Type drop-down list in the Identity Access Control List Configuration section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

8. Click Search.

   A list of application groups is displayed.

9. Select the application group you created, and click  .

   The application group appears in the Selected Identities list.

10. Enter local/* in the Add Resource field in the Resources section, and click  .

    The resource is added to the resource list, providing access to the local command list (My Commands tab) in the Command Line Console application.

11. Enter global/* in the Add Resource field, and click  .

    The resource is added to the resource list, providing access to the global command list (Global Commands tab) in the Command Line Console application.

12. Select the Create, Modify, and View check boxes under Actions for your group.

13. Click Save at the top or bottom right of the page.

    The policy is created.

14. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create a CommandExecute Policy

To enable a user to execute commands in the Enterprise Command Line application (or Command Line Console), you can create a CommandExecute policy. For this policy, a type of Access Policy is recommended. This example is written based on that selection and assumes that the users in the group will have rights to execute commands on all servers to which they have access.

The default policy, CommandExecuteDefault, lets users in the Administrator, Console Operator, Scheduler, Supervisor, and Commander (superuser) groups perform commands on the servers identified in the resource class.

To create a CommandExecute policy for your application group

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click CommandExecute in the left pane.

   All the CommandExecute policies are listed in the right pane.

3. Click the New Access Policy icon next to CommandExecute in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, leave CommandExecute in the Resource Class Name drop-down list, and leave the Access Policy option button selected in the Type field.

5. Select Application Group from the Type drop-down list in the Identities section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

6. Click Search.

   A list of application groups is displayed.

7. Select the application group you created, and click .

   The application group appears in the Selected Identities box to the right.

8. Enter server/* in the Add Resource field in the Resources section, and click .

   The resource is added to the resource list.

9. Select the Execute check box under Actions in the Access Policy Configuration section for your group.

10. Click Save at the top or bottom right of the page.

    The policy is created.

11. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Create a ConfigurationControl Policy

To control various settings for the configuration-related applications, you can create a ConfigurationControl policy. The ConfigurationControl policy can be used to control the configuration and operation of several disparate resources. Currently, you can create policies for configuring the CPM (Critical Path Monitoring), Credentials, Framework, Monitoring, and High Availability applications. For this policy, a type of Access Policy is recommended. This example is written based on that selection. In this example, we create a policy to control the CPM application.

Note: It is recommended that you create a separate policy for each type of application to be configured.

The default policies, ConfigurationControlCPMDefault, ConfigurationControlCredentialsAdminDefault, ConfigurationControlCredentialsUserDefault, ConfigurationControlHighAvailabilityDefault, ConfigurationControlFrameworkThemesDefault, ConfigurationControlMonitoringAdminDefault, and ConfigurationControlMonitoringUserDefault, establish different access levels for the CPM, Credentials, High Availability, Framework, and Monitoring applications respectively. For example, only the Commander (superuser), Console Operator, Scheduler, and Supervisor groups can configure access to the CPM application; whereas, only the Commander and Administrator groups can configure access to the Credentials application (although all users can modify their own credentials) and the High Availability application.

To create a ConfigurationControl policy

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default. In the left pane, the Show policies matching name option is selected by default.

2. (Optional) Click ConfigurationControl in the left pane.

   All the ConfigurationControl policies are listed in the right pane.

3. Click the New Access Policy icon  next to ConfigurationControl in the left pane.

   The New Access Policy page appears in the right pane.

4. Enter a unique policy name in the Name field, leave ConfigurationControl in the Resource Class Name drop-down list, and leave the Access Policy option button selected in the Type field.

5. Select Application Group from the Type drop-down list in the Identities section, and click Search Identities.

   The Attribute, Operator, and Value fields appear.

6. Click Search.

   A list of application groups is displayed.

7. Select the application group you created, and click  .

   The group appears in the Selected Identities box to the right.

8. Enter CPM/MyFilter in the Add resource field in the Access Policy Configuration section, and click  .

   The resource is added to the resource list, providing access to personal filters.

9. Select the Save check box under Actions for this resource.

   The ability to modify and save personal filters is enabled for users associated with the selected group.

10. Click Save at the top or bottom right of the page.

    The policy is created.

11. Perform a permission check as appropriate.

More information:

Log in to CA EEM for CA WCC (see page 128)
Perform a Permission Check (see page 150)

Perform a Permission Check

To check whether access permission has been properly assigned, you can perform a permission check.

To perform a permission check

1. Select the Manage Access Policies tab in the CA EEM application.

   The Policies, Calendars, and Permission Check links appear, with the Policies link selected by default.

2. Click the Permission Check link.

   The Permission Check page opens.

3. Click Synchronize Cache.

   A confirmation message appears.

4. Select the appropriate policy from the Resource Class drop-down list in the Permission Check Parameters section.

5. Select the action for the respective policy from the Action drop-down list as follows:

Policy	Action
ApplicationAccess	Framework
ServerAccess	Access
ObjectControl	Modify
ObjectAccess	Access
MonitorViewControl	View
LogAccess	JobLog
AlertAction	Acknowledge
JobActionAutoSys	Start
CommandSetup	View
CommandExecute	Execute
ConfigurationControl - CPM	Save

6. Enter a value for the respective policy in the Resource field as follows:

Policy	Resource
ApplicationAccess	*
ServerAccess	server/SrvrAutoSys
ObjectControl	*/Forecast/*
ObjectAccess	*/AlertPolicy/*
MonitorViewControl	view/*viewName*/JobFlow
LogAccess	server/SrvrAutoSys
AlertAction	view/*viewName*
JobActionAutoSys	view/*viewName*
CommandSetup	global/*
CommandExecute	server/*
ConfigurationControl - CPM	CPM/MyFilter

7. Enter the user ID you created in the Identity field.
8. Click Run Permission Check.

   The results of the permission check appear in the Permission Check Results section. If correct, this check should return a result of Allow.

More information:

Log in to CA EEM for CA WCC (see page 128)

Verify the Access Set by Your Policies

After you have completed the policy creation and permission checks, log in to CA WCC as a user in your application group and verify that all of the policies you added are working as designed.

Chapter 7: CA Workload Automation AE Policy Migration

This chapter describes how to migrate your security policy from CA AC to CA EEM.

This section contains the following topics:

Requirements to Migrate from Unicenter AutoSys JM 4.5 or 4.5.1

The CA EEM policy migration requires the following files:

- antl.jar
- se2xml.jar
- AutoSys.xsl
- PostRegex.xsl
- selang2eem.xsl

Note: These files are included with the CA Workload Automation AE installation and are located in the $AUTOSYS/EEMmigrate directory after CA Workload Automation AE is installed.

The following tools are used to migrate your existing CA AC policy to CA EEM:

CA Workload Automation AE as_safetool Utility

Installs the default policies for all the instances that have CA AC security policies associated with them. CA Workload Automation AE installs the as_safetool utility.

CA EEM safex Utility

Imports the final generated XML file containing the migrated policies to the CA EEM back-end server.

Note: The safex utility is only available on a computer where the CA EEM back-end server is installed. This utility is located in the iTechnology subdirectory of the SharedComponents directory.

Java Runtime Environment (JRE)

Runs the Java commands that are part of the migration policy.

Note: The JRE is located in the SharedComponents directory. Make sure the PATH environment variable is updated to include the location of the java binary.

Security Policy Changes from Unicenter AutoSys JM 4.5 or 4.5.1

This section describes the changes to the security policy implementation in the current release.

Before the CA AC security policy implementation can be successfully imported into CA EEM, you must apply these changes as part of the migration process.

Deprecated Security Classes and Resources

The following security classes and resources are deprecated in this release:

- The as-view resource class has been deprecated and is not imported.
- The following as-control resources have been deprecated and are not imported:
  - Resources ending with _ON, _OFF
  - Resources beginning with WEBADM
- The following as-list resources have been deprecated and are not imported:
  - Resources beginning with AUTOCONS
  - Resources beginning with JOBDEF
  - Resources beginning with XPERT

CA AC Default Resource

In Unicenter AutoSys JM 4.5 and 4.5.1, every CA AC resource class contains a default resource (with the name _default) defining the security policy for objects that do not have a matching policy.

In the current release, the default resource does not exist. Therefore, the migration process converts the CA AC default resources to an equivalent CA EEM policy for objects with no matching policies.

Resource Naming Convention

In Unicenter AutoSys JM 4.5 and 4.5.1, CA AC resource names that were created for all resource classes (except as-owner) followed this naming convention, the protected object name followed by a period followed by the Unicenter AutoSys JM instance (*object.instance*).

In the current release, the order of the protected object name and CA Workload Automation AE instance in the CA EEM resource name has been reversed (*instance.object*). Therefore, the migration process applies the following conversion rules to the resource names in all classes, except as-owner:

- *object*.*instance* becomes *instance*.*object*
- *object*.* becomes *.*object*
- *object** becomes *.*object**

Note: The migration process relies on the CA AC resource name following the naming convention *object*.*instance* to detect the CA Workload Automation AE instance. For example, the migration process will not properly convert a CA AC resource name with the naming convention *object***instance* with no period.

Asterisks in Resource Names

Resource names in CA AC may contain multiple asterisks (*) to form simple regular expressions.

By default, CA EEM can only interpret asterisks in resource names if they are located in the first or last position. Asterisks in positions other than the first or last character are treated literally and not as special characters. For a resource name containing additional asterisks to be treated as a regular expression, CA EEM requires that the policy's regular expression attribute be set. Policies with the regular expression attribute support simple regular expressions with syntax and semantics similar to the Perl 5 language. The final step of the migration process scans the converted CA EEM policies for resource names containing asterisks in positions other than first or last. If such a policy is found, the regular expression attribute of the policy is set and every asterisk in the resource name is prefixed with a period to conform to a Perl 5 regular expression. Therefore, the migration process applies the following conversion rules to the resource names to convert the CA AC resource names to their CA EEM equivalents:

- *[*object*] remains unchanged
- [*object*]* remains unchanged

- *[*object*]* remains unchanged
- [*object*]*[*object*] becomes regular expression policy [*object*].*[*object*]
- *[*object*]*[*object*] becomes regular expression policy .*[*object*].*[*object*]
- [*object*]*[*object*]* becomes regular expression policy [*object*].*[*object*].*
- *[*object*]*[*object*]* becomes regular expression policy .*[*object*].*[*object*].*

How to Migrate Security Policies from CA AC to CA EEM

Typically, a security policy definition in an enterprise consists of the following:

- User and group definitions
- Access level definitions of resources for the users and groups

Therefore, to migrate security policies from CA AC to CA EEM, you must do the following:

1. Migrate all users and groups from CA AC to CA EEM (see page 158).
2. (Optional) Migrate global users and groups from CA AC to CA EEM (see page 159).
3. Migrate individual resource policies from CA AC to CA EEM (see page 160).

Note: The migration of all users and groups must take place independently of the migration of the individual policies.

How to Migrate Users and Groups from CA AC to CA EEM

In CA AC, users and groups are represented by a script file containing specific commands written in selang, the CA AC command language. In CA EEM, users and groups are represented by an XML file using specific CA EEM XML tags. The migration process involves obtaining the CA AC users and groups from the selang file and translating them to an XML file containing equivalent CA EEM users and groups. The resulting XML file is imported to the CA EEM back-end server.

Note: For this process to be successful, you must make sure the PATH environment variable is updated to include the location of the Java binary.

To migrate users and groups from CA AC to CA EEM, do the following:

1. Export CA AC users and groups to a selang file (see page 161).
2. Convert the selang file to a selang XML file (see page 162).
3. Convert the selang XML file to a CA EEM XML file (see page 166).
4. Import the final CA EEM XML file to the CA EEM back-end server (see page 168).
5. Clean up files (see page 168).

How to Migrate Global Users and Groups from CA AC to CA EEM

If you use global users and user groups in your enterprise, you can migrate them to CA EEM. Global users and user groups can be shared across all application instances registered with CA EEM.

In CA AC, users and groups are represented by a script file containing specific commands written in selang, the CA AC command language. In CA EEM, users and groups are represented by an XML file using specific CA EEM XML tags. Users and groups are defined differently in CA AC and CA EEM. Therefore, the migration process involves getting the CA AC users and groups from the selang file and translating them to an XML file containing equivalent CA EEM users and groups. When you migrate global users and user groups to CA EEM, you must create the XML file manually. That XML file is imported to the CA EEM back-end server.

Note: You must ensure that the PATH environment variable is updated to include the location of the Java binary.

To migrate global users and groups from CA AC to CA EEM, do the following:

1. Export CA AC users and groups to a selang file (see page 161).
2. Convert the selang file to a selang XML file (see page 162).
3. Manually create a CA EEM XML file for global users and groups from the selang XML file (see page 163).
4. Import the final CA EEM XML file to the CA EEM back-end server (see page 168).
5. Clean up the files (see page 168).

How to Migrate Security Policies from CA AC to CA EEM

In CA AC, policies are represented by a script file containing specific commands written in selang, the CA AC command language. In CA EEM, policies are represented by an XML file using specific CA EEM XML tags. The migration process involves obtaining the subset of CA AC security policies used by Unicenter AutoSys JM 4.5 and 4.5.1 and translating them to an XML file containing equivalent policies for the current release. The resulting XML file is imported to the CA EEM back-end server. These steps are necessary because of the differences in the policy evaluation of the two security engines and changes in the resource naming convention between Unicenter AutoSys JM 4.5 and 4.5.1 and the current release.

Note: For this process to be successful, you must make sure the PATH environment variable is updated to include the location of the Java binary.

To migrate security policies from CA AC to CA EEM, do the following:

1. Register CA Workload Automation AE instances with the CA EEM back-end server (see page 161).
2. Export CA AC policies to a selang file (see page 162).
3. Convert the selang file to a selang XML file (see page 162).
4. Convert the selang XML file to a CA EEM XML file (see page 166).
5. Apply security policy changes to Unicenter AutoSys JM 4.5 or 4.5.1 policies (see page 167).
6. Apply regular expression resource name changes to policies for the current release (see page 167).
7. Import the final CA EEM XML file to the CA EEM back-end server (see page 168).
8. Clean up files (see page 168).

Migration Procedures

The following procedures must be performed to migrate users, groups, and resource policies.

Register CA Workload Automation AE Instances with the CA EEM Back-end Server

The CA Workload Automation AE as_safetool command is used to register CA Workload Automation AE instances with the CA EEM back-end server. You must individually register the instance names that are represented in the CA AC policies with the CA EEM back-end server.

Note: Before performing this task, you must use the setenv or export command to set the ASSAFETOOLPW environment variable to the password of the EiamAdmin user, the back-end server administrative account. For example, export ASSAFETOOLPW=mypass.

To register CA Workload Automation AE instances with the CA EEM back-end server

1. Open a CA Workload Automation AE command prompt.
2. Enter the following command:

   `as_safetool -b EEM_backend_server_host_name -s`

   A list of CA Workload Automation AE instances that are already registered with the CA EEM back-end server appear.
3. Enter the following command for each instance that is represented in the CA AC policy, but is not part of the list derived from the previous step:
   `as_safetool -b EEM_backend_server_host_name -i instance`

   The CA Workload Automation AE instance is registered with the CA EEM back-end server.

   Note: The as_safetool command installs some default CA EEM policies for each CA Workload Automation AE instance. We recommend that you review these policies and update them accordingly.

Export CA AC Users and Groups to a selang File

CA AC provides the dbmgr utility to export the necessary users and groups into a script file containing the selang commands required to duplicate the database.

Note: For more information about how to use the dbmgr utility to export users and groups into a selang file, see the *CA Access Control Reference Guide*.

Export CA AC Policies to a selang File

After registering the instances with the CA EEM back-end server, you must export all of the Unicenter AutoSys JM 4.5 or 4.5.1 resources from CA AC into a script file containing the selang commands required to duplicate the database. You must export resources only from the following user-defined classes:

- as-calendar
- as-control
- as-cycle
- as-gvar
- as-job
- as-list
- as-machine
- as-owner

CA AC provides the dbmgr utility to export the necessary resources into a script file containing the selang commands required to duplicate the database.

Note: For more information about how to use the dbmgr utility to export resources from the listed classes into a selang file, see the *CA Access Control Reference Guide*.

Convert the selang File to a selang XML File

The JRE is required to convert the selang file to a selang XML file. You must identify the selang commands from the exported file and generate an equivalent XML file containing the selang commands as XML tags. After the script file is converted to an XML file, you can use an XML parser to translate the CA AC XML tags to the equivalent CA EEM XML tags.

To convert the selang file to a selang XML file, go to the $AUTOSYS/EEMmigrate directory and enter the following command:

```
java -jar se2xml.jar exported_selang_file_name
```

exported_selang_file_name

Identifies the name of the selang file to convert.

An XML file with the name *exported_selang_file_name*.xml is generated.

Manually Create a CA EEM XML File for Global Users and Groups from the selang XML File

To migrate global users and groups from CA AC to CA EEM, you must generate one or more XML files containing the users and user groups defined in the selang XML tags.

To manually create a CA EEM XML file for global users and groups from the selang XML file

1. Create an CA EEM XML file.
2. Add the following lines to the file:

```
<Safex>
<Attach />
<Add>
```

3. To define a global user, add the following lines:

```
<GlobalUser folder="root_path_to_CA_EEM_server" name="user_name">
<user_attribute>attribute_value</user_attribute>
<user_attribute>attribute_value</user_attribute>
<user_attribute>attribute_value</user_attribute>
...
</GlobalUser>
```

user_attribute

Specifies an attribute for the user. Options are the following:

- UserName
- GroupMembership
- FirstName
- MiddleName
- LastName
- EmailAddress
- Alias
- Department
- DisplayName
- HomePhoneNumber
- WorkPhoneNumber
- MobilePhoneNumber
- FaxPhoneNumber
- Address
- City

- State
- PostalCode
- Country
- Office
- Company
- PasswordDigest
- IncorrectLoginCount
- SuspendDate
- DisableDate
- EnableDate
- Description
- Comments
- JobTitle
- MailStop

4. Repeat Step 3 for each additional global user you want to define.
5. To define a global user group, add the following lines:

```
<GlobalUserGroup folder="root_path_to_CA_EEM_server" name="group_name">
<group_attribute>attribute_value</group_attribute>
<group_attribute>attribute_value</group_attribute>
<group_attribute>attribute_value</group_attribute>
...
</GlobalUserGroup>
```

group_attribute

Specifies an attribute for the group. Options are the following:

- GroupMembership
- Description

6. Repeat Step 5 to for each global user group you want to define.
7. Add the following lines to the end of the file:

```
</Add>
</Safex>
```

8. Save the XML file.

The CA EEM XML file is created.

Example: Create an XML File for a Global User

This example defines a user XML file for global user "johndoe". This file contains all the user attributes that you can use.

```
<Safex>
<Attach />
<Add>
<GlobalUser folder="/" name="johndoe">
<UserName>doejo33</UserName>
<GroupMembership>Administrators</GroupMembership>
<FirstName>john</FirstName>
<MiddleName>dennis</MiddleName>
<LastName>doe</LastName>
<EmailAddress>jdoe@example.com</EmailAddress>
<Alias>jdoe</Alias>
<Department>accounting</Department>
<DisplayName>John D Doe</DisplayName>
<HomePhoneNumber>718-264-8966</HomePhoneNumber>
<WorkPhoneNumber>508-628-7076</WorkPhoneNumber>
<MobilePhoneNumber>508-593-0963</MobilePhoneNumber>
<FaxPhoneNumber>508-628-2319</FaxPhoneNumber>
<Address>331 Main St</Address>
<Address>Jones Building</Address>
<Address>Suite 3200</Address>
<Address>Acme Corp.</Address>
<City>Smallville</City>
<State>Quebec</State>
<PostalCode>H4M2X4</PostalCode>
<Country>Canada</Country>
<Office>C-42</Office>
<Company>Acme</Company>
<PasswordDigest>xxxxxxxxxxxxxx</PasswordDigest>
<IncorrectLoginCount>0</IncorrectLoginCount>
<SuspendDate>0</SuspendDate>
<DisableDate>0</DisableDate>
<EnableDate>0</EnableDate>
<Description>Working in Finance</Description>
<Comments>12 month temp</Comments>
<JobTitle>Billing Manaeger</JobTitle>
<MailStop>C-42-2-12</MailStop>
</GlobalUser>
</Add>
</Safex>
```

Example: Create an XML File for Global User Groups

This example defines a user group XML file for global user groups "Staff" and "Administrator". This file contains all the user group attributes that you can use.

```
<Safex>
<Attach />
<Add>
<GlobalUserGroup folder="/" name="Staff">
<Description>Staff group description</Description>
</GlobalUserGroup>
<GlobalUserGroup folder="/" name="Administrators">
<GroupMembership>Staff</GroupMembership>
<Description>Administrator group description</Description>
</GlobalUserGroup>
</Add>
</Safex>
```

Convert the selang XML File to a CA EEM XML File

The JRE is required to convert the selang XML file to a CA EEM XML file. An XML parser is used to identify the CA AC tags from the selang XML file and generate an XML file containing the equivalent CA EEM tags.

To convert the selang XML file to a Unicenter AutoSys JM 4.5 or 4.5.1 CA EEM XML file, go to the $AUTOSYS/EEMmigrate directory and enter the following command:

```
java org.apache.xalan.xslt.Process -IN exported_selang_file_name.xml
-XSL selang2eem.xsl -OUT EEM_file_name.xml
-PARAM ApplicationName WorkloadAutomationAE
-PARAM PoliciesFolder EEM_backend_server_policy_folder_name
```

***exported_selang_file_name*.xml**

Identifies the selang XML file to convert.

***EEM_file_name*.xml**

Defines the name of the CA EEM XML file to create.

EEM_backend_server_policy_folder_name

Specifies the path on the CA EEM back-end server where the policies will be imported.

Limits: You must precede this value with a slash. For example, you could use /MigratedPolicies.

An XML file with the name *EEM_file_name*.xml is generated.

Apply Security Policy Changes to Unicenter AutoSys JM 4.5 or 4.5.1 Policies

The JRE is required to convert the Unicenter AutoSys JM 4.5 or 4.5.1 CA EEM XML file to a CA EEM XML file for the current release. You must apply the security policy changes required to work with the current release.

To apply security policy changes to Unicenter AutoSys JM 4.5 or 4.5.1 policies in the CA EEM XML file, go to the $AUTOSYS/EEMmigrate directory and enter the following command:

```
java -Xmx128M org.apache.xalan.xslt.Process -IN EEM_file_name.xml
-XSL AutoSys.xsl -OUT WAAE_EEM_file_name.xml
```

***EEM_file_name*.xml**

Identifies the CA EEM XML file generated from the selang file.

***WAAE_EEM_file_name*.xml**

Defines the name of the CA EEM XML file to create. This file contains the policy changes for the current release.

An XML file with the name *WAAE_EEM_file_name*.xml is generated.

Apply Regular Expression Resource Name Changes to Policies for the Current Release

The JRE is required to convert the Unicenter AutoSys JM r11 CA EEM XML file to a CA EEM XML file with regular expression policies for the current release. The purpose of this migration step is to scan the converted CA EEM policies for any resource names containing asterisks in positions other than first or last. If a policy with these asterisks is found, the regular expression attribute of the policy is set and every asterisk in the resource name is prefixed with a period to conform to a Perl 5 regular expression.

To apply regular expression resource name changes to policies for the current release in the CA EEM XML file, go to the $AUTOSYS/EEMmigrate directory and enter the following command:

```
java -Xmx128M org.apache.xalan.xslt.Process -IN WAAE_EEM_file_name.xml
-XSL PostRegex.xsl -OUT final_EEM_file_name.xml
```

***WAAE_EEM_file_name*.xml**

Identifies the CA EEM XML file to modify.

***final_EEM_file_name*.xml**

Defines the name of the final CA EEM XML file to create.

An XML file with the name *final_EEM_file_name*.xml is generated.

Import the Final CA EEM XML File to the CA EEM Back-end Server

The CA EEM safex utility is required to import the final CA EEM XML file to the CA EEM back-end server. This utility is only available on the computer acting as your CA EEM back-end server.

The final CA EEM XML file, created either by migrating the users and groups or the resource policies, represents the completed conversion from Unicenter AutoSys JM 4.5 or 4.5.1 policies in the selang command language to policies in CA EEM XML for the current release. The final step in the migration process is to import this XML file to the CA EEM back-end server. This adds the security policies to the appropriate repository for use by the current release.

To import the current release CA EEM XML file to the CA EEM back-end server, go to the iTechnology subdirectory of the SharedComponents directory, which is located at the same level as the CA Workload Automation AE installation path and enter the following command:

```
safex -u EiamAdmin -p EiamAdmin_account_password
-f final_EEM_file_name.xml
```

EiamAdmin_account_password

Specifies the password of the EiamAdmin user, the back-end server administrative account with permissions to update the CA EEM back-end server.

***final_EEM_file_name*.xml**

Identifies the final CA EEM XML file.

Note: The safex utility directs all output to stderr. We recommend that you capture this output and save it to a file so you can examine errors.

The converted CA AC policies are imported to the WorkloadAutomationAE application instance on the CA EEM back-end server.

Clean Up Files

After you have finished migrating CA AC users, groups, and policies to CA EEM, to clean up, remove the selang file and all the intermediate XML files created during the process.

How to Migrate Security Policies from Unicenter AutoSys JM r11 to the Current Release

In CA EEM, security policies are represented by an XML file using specific CA EEM XML tags. To migrate security policies from Unicenter AutoSys JM r11 to the current release, do the following:

- Export Unicenter AutoSys JM r11 security policies to an XML file (see page 169).
- Modify the XML file to match the current release security policies (see page 172).
- Import the modified XML file to CA EEM (see page 174).
- Clean up files (see page 174).

These steps are necessary because of the following changes in the security policies between Unicenter AutoSys JM r11 and the current release:

- The application instance name is changed from UnicenterAutoSysJM to WorkloadAutomationAE.
- A new resource class named *as-resource* is added to support the authorization of resources.
- The READ and WRITE access modes of the as-group and as-appl resource classes are updated.

Note: You must migrate the security policies created in Unicenter AutoSys JM r11 to the current release before you activate CA EEM for the CA Workload Automation AE instance.

Export Unicenter AutoSys JM r11 Security Policies to an XML File

You can export the Unicenter AutoSys JM r11 security policies to an XML file using any *one* of the following methods:

- Export the Unicenter AutoSys JM r11 security policies to an XML file using the CA EEM web interface (see page 170).
- Export the Unicenter AutoSys JM r11 security policies to an XML file using the safex utility (see page 171).

Export the Unicenter AutoSys JM r11 Security Policies to an XML File Using CA EEM

You can export the Unicenter AutoSys JM r11 security policies to an XML file using the CA EEM web interface.

To export Unicenter AutoSys JM r11 security policies to an XML file using CA EEM

1. Open a browser and go to the following web site:

   `http:\\localhost:5250\spin\eiam`

   localhost

   Specifies the IP address or host name of the computer where CA EEM is installed.

   The CA EEM login page appears.

2. Select WorkloadAutomationAE instance from the Application drop-down list, enter EiamAdmin in the User Name field and the appropriate password in the Password field, and click Log In.

   The CA EEM web interface opens. The Home tab is displayed by default.

3. Select the Configure tab.

   The Applications, Folders, Session, and EEM Server links appear, with the Applications link selected by default.

4. Select the EEM Server link.

   In the left pane, the Global Users/Global Groups, EiamAdmin Password, Password Policies, Cached Events, Configure SAF Location, Export Application, PassTicket Configuration, and Artifact Authentication links appear.

5. Select the Export Application link.

   The Export Application dialog appears, displaying the Object List check boxes.

6. Select the check boxes as appropriate, and click Export.

   The File Download dialog appears.

7. Specify the file name and location to save the XML file in zip format.

8. Extract the zip file.

   The XML file is extracted. All the security policies are retrieved from the UnicenterAutoSysJM application instance, and this XML file contains the policies for all CA Workload Automation AE instances.

Export the Unicenter AutoSys JM r11 Security Policies to an XML File Using the Safex Utility

You can use the safex utility to export the Unicenter AutoSys JM r11 security policies to an XML file.

Note: The safex utility is only available on the computer acting as your CA EEM back-end server.

To export Unicenter AutoSys JM r11 security policies to an XML file, go to the iTechnology subdirectory of the SharedComponents directory, which is located at the same level as the CA Workload Automation AE installation and enter the following command:

```
safex –h EEM_servername –u EiamAdmin –p EiamAdmin_account_password –f
input_safex_file_name.xml
```

EEM_servername

Defines the name of your CA EEM back-end server.

EiamAdmin_account_password

Specifies the password of the EiamAdmin user, the back-end server administrative account with permissions to update the CA EEM back-end server.

***input_safex_file_name*.xml**

Identifies the XML file that contains the commands to export Unicenter AutoSys JM r11 policies.

The content of the input safex XML file is similar to the following:

```
<Safex>
<Attach label="UnicenterAutoSysJM"/>
<Export file="Path for export file" globalfolders="y" globalusergroups="y"
globalusers="y" globalsettings="y" folders="y" usergroups="y" users="y"
calendars="y" policies="y" appobjects="y"/>
<Detach/>
</Safex>
```

The XML file is created in the location specified by the *Export file* tag of the input safex XML file. All the security policies are retrieved from the UnicenterAutoSysJM application instance, and this XML file contains the policies for all CA Workload Automation AE instances

Modify the XML File to Match the Current Release Security Policies

You must modify the XML file to match the current release security policies to apply the security policy changes required to work with the current release.

To modify the XML file to match the current release security policies

1. Open the XML file in a text editor.
2. Search for the ApplicationInstance tag and make the following changes:
   a. In the name and label attributes, replace Unicenter AutoSys JM with Workload Automation AE and UnicenterAutoSysJM with WorkloadAutomationAE.
   b. Change the value of the MinorVersion tag to 3.
   c. In the Translation tag, do the following:
      - Replace Unicenter AutoSys JM with Workload Automation AE and UnicenterAutoSysJM with WorkloadAutomationAE.
      - Add the following tag after the as-owner string tag:

        ```
        <string> <key>as-resource</key> </string>
        ```

   d. In the Resources tag, search for as-appl and as-group tags, and add the following tag:

      ```
      <Action>read</Action>
      <Action>write</Action>
      ```

   e. Add the ResourceClass tag as follows:

      ```
      <ResourceClass>
       <Name>as-resource</Name>
       <BestMatchEvaluation>true</BestMatchEvaluation>
       <Action>read</Action>
       <Action>write</Action>
       <Action>create</Action>
       <Action>delete</Action>
       <Action>execute</Action>
      </ResourceClass>
      ```

   f. Search for the FolderName tag and replace UnicenterAutoSysJM_Common with WorkloadAutomationAE_Common and UnicenterAutoSysJM_XXX with WorkloadAutomationAE_XXX.
   g. Search for the UserGroup tag and replace UnicenterAutoSysJMAdmin with WorkloadAutomationAEAdmin.
   h. For each Policy tag, replace /UnicenterAutoSysJM_*XXX* with /WorkloadAutomation_*XXX,* where *XXX* is the CA Workload Automation AE instance name.

i. Add default policies for the as-resource resource class for each CA Workload Automation AE instance by replacing *XXX* with the CA Workload Automation AE instance name in the following tags:

```
<Policy folder="/WorkloadAutomationAE_XXX" name="XXX: Default Resource
Policy">
        <ResourceClassName>as-resource</ResourceClassName>
        <PolicyType>identityacl</PolicyType>
        <Disabled>False</Disabled>
        <ExplicitDeny>False</ExplicitDeny>
        <PreDeployment>False</PreDeployment>
        <RegexCompare>False</RegexCompare>
        <Resource>XXX.*</Resource>
<Action>read</Action>
<Action>write</Action>
        <Action>execute</Action>
        <Attribute name="CreateTimestamp">20090514052523</Attribute>
</Policy>
```

j. Search for UnicenterAutoSysJM and replace with WorkloadAutomationAE.

k. Modify all the policies for as-appl and as-group resource classes by adding the following tags:

```
<Action>read</Action>
<Action>write</Action>
```

Note: You must add the tags to each corresponding CA Workload Automation AE instance policy as follows:

```
<Policy folder="/WorkloadAutomationAE_XXX" name="XXX: Default Groups
Policy">
        <ResourceClassName>as-group</ResourceClassName>
        <PolicyType>identityacl</PolicyType>
        <Disabled>False</Disabled>
        <ExplicitDeny>False</ExplicitDeny>
        <PreDeployment>False</PreDeployment>
        <RegexCompare>False</RegexCompare>
                <Resource>XXX.*</Resource>
                <Action>read</Action>
                <Action>write</Action>
                <Action>execute</Action>
</Policy>
```

The XML file is modified to match the current release security policies.

Import the Modified XML File to CA EEM

The safex utility is used to import the modified XML file to CA EEM.

Note: The safex utility is only available on the computer acting as your CA EEM back-end server.

To import the modified XML file to CA EEM, go to the iTechnology subdirectory of the SharedComponents directory, which is located at the same level as the CA Workload Automation AE installation and enter the following command:

`safex –h EEM_servername –u EiamAdmin –p EiamAdmin_account_password –f modified_EEM_file_name.xml`

EEM_servername

Defines the name of your CA EEM back-end server.

EiamAdmin_account_password

Specifies the password of the EiamAdmin user, the back-end server administrative account with permissions to update the CA EEM back-end server.

***modified_EEM_file_name*.xml**

Identifies the modified XML file.

Clean Up Files

After you have finished migrating security policies from Unicenter AutoSys JM r11 to the current release, to clean up, remove the intermediate XML files created during the process.

Chapter 8: CA EEM Data Replication/Backup for CA WCC

This chapter includes procedures for backing up and restoring the CA EEM policies, users, and user group information created for CA WCC.

This section contains the following topics:

Configure Data Store Replication Using Multi-Write

You can use the CA EEM multi-write feature to automatically synchronize data on the CA EEM instances. Although this feature enables continual and automatic synchronization, its disadvantages include replication of unintentional changes, and no central data backup in case of catastrophic failure.

The knowledge files provide the server information needed to set up data store replication.

You must configure the following knowledge files:

- Data knowledge file (iTechPoz-HostnameOfServerN.dxc) to add the host name of each server
- Router knowledge file (iTechPoz-HostnameOfServerN-Router.dxc) to add the host information for the router
- Group knowledge file (iTechPoz.dxg) to add group knowledge that all Directory System Agents (DSAs) in the domain can access

To configure knowledge files

1. Open the knowledge directory, as follows:
   - **Windows:** Open a command prompt and enter %DXHOME%\config\knowledge.
   - **UNIX:** Open a command prompt and enter ~dsa/config/knowledge.
2. Modify the Server1 router knowledge file (iTechPoz-HostnameOfServer1-Router.dxc) as follows:
   a. Change the following entry from:

      ```
      dsa-name = <cn iTechPozRouter><cn PozDsa>
      ```

      to:

      ```
      dsa-name = <cn iTechPozRouter><cn PozDsaHostnameOfServer1>
      ```

   b. Add the following entry after the auth-levels line and before the link-flags line:

      ```
      dsa-flags = multi-write
      ```

   c. Save and close the Server1 router knowledge file.
3. Copy the Server1 router knowledge file to all the servers in the failover setup, and rename the copied file to iTechPoz-HostnameOfServerN-Router.dxc.
4. Modify the Server2 router knowledge file as follows:
   a. Change the following entries from:

      ```
      set dsa "iTechPoz-Server1-Router"
      ```

```
dsa-name = <cn iTechPozRouter><cn PozDsaHostnameOfServer1>
```

to:

```
set dsa "iTechPoz-Hostname_ServerN-Router"

dsa-name = <cn iTechPozRouter><cn PozDsaHostnameOfServerN>
```

5. Modify the Server1 knowledge file (iTechPoz-HostnameOfServer1.dxc) and set the following preferences:

   a. Change the following entries from:

   ```
   tcp localhost port 509

   dsa-name = <cn iTechPoz><cn PozDsa>
   ```

   to:

   ```
   tcp HostnameOfServer1 port 509, tcp localhost port 509

   dsa-name = <cn iTechPoz><cn PozDsaHostnameOfServer1>
   ```

   b. Add the following entry after the auth-levels line and before the link-flags line, if the entry dsp-idle-time is not present in the knowledge file:

   ```
   dsa-flags = multi-write
   ```

   Note: If the entry dsp-idle-time is present in the knowledge file, add the dsa-flags entry between the dsp-idle-time and link-flags entries.

6. Copy the Server1 knowledge file to all the servers in the failover setup, and rename the copied file to iTechPoz-HostnameOfServerN.dxc.

7. Modify the ServerN knowledge file (iTechPoz-HostnameOfServerN.dxc) as follows:

   a. Change the following entries from:

   ```
   set dsa "iTechPoz-Server1"

   tcp HostnameOfServer1 port 509, tcp localhost port 509

   dsa-name = <cn iTechPoz><cn PozDsaHostnameOfServer1>
   ```

   to:

   ```
   set dsa "iTechPoz-Server2"

   tcp HostnameOfServerN port 509, tcp localhost port 509

   dsa-name = <cn iTechPoz><cn PozDsaHostnameOfServerN>
   ```

   b. Add the following entry after the auth-levels line and before the link-flags line, if the entry dsp-idle-time is not present in the knowledge file:

   ```
   dsa-flags = multi-write
   ```

   Note: If the entry dsp-idle-time is present in the knowledge file, add the dsa-flags entry between the dsp-idle-time and link-flags entries.

8. Modify the group knowledge file (iTechPoz.dxg) and set the preferences to the new data and router knowledge files on Server1 and all other servers in the failover setup as follows:

   a. Change the iTechPoz.dxg file on Server1 to:

```
# iTechPoz - iTechnology rePOZitory

# Source the knowledge files of the iTechPozRouter and iTechPoz DSAs.

source "iTechPoz-HostnameofServer1-Router.dxc";

source "iTechPoz-HostnameofServer1.dxc";

source "iTechPoz-HostnameOfServer2-Router.dxc";

source "iTechPoz-HostnameOfServer2.dxc";

.

.

.

.

source "iTechPoz-ServerN-Router.dxc";

source "iTechPoz-ServerN.dxc";
```

   b. Change the iTechPoz.dxg file on ServerN to:

```
# iTechPoz - iTechnology rePOZitory

# Source the knowledge files of the iTechPozRouter and iTechPoz DSAs.

source "iTechPoz-HostnameOfServerN-Router.dxc";

source "iTechPoz-HostnameOfServerN.dxc";

source "iTechPoz-HostnameOfServer1-Router.dxc";

source "iTechPoz-HostnameOfServer1.dxc";

source "iTechPoz-HostnameOfServer2-Router.dxc";

source "iTechPoz-HostnameOfServer2.dxc";

.

.

.

source "iTechPoz-ServerKRouter.dxc";

source "iTechPoz-ServerK.dxc";
```

9. (Optional) Modify the owner of the following files to dsa and owner of the group files to etrdir for CA EEM for all the servers, in the failover setup that are running on UNIX:

   - iTechPoz-HostnameofServer1-Router.dxc

- iTechPoz-HostnameOfServerN-Rotuer.dxc
- iTechPoz-HostnameOfServer1.dxc
- iTechPoz-HostnameOfServerN.dxc

10. Copy the certificate files of Server1 to all the servers in the failover setup. Similarly, copy the certificate files from each of the servers to every other server in the failover setup as follows:

    Note: Certificate files are found in the %DXHOME%\config\ssld\personalities directory.

    a. Copy the file itechpoz-HostnameOfServer1-router.pem from Server1 to Server2, Server 3, ..., and Server N

    b. Copy the file itechpoz-HostnameOfServer2-router.pem from Server2 to Server1, Server2, Server3,....., and ServerN

11. Create a new iTechPoz-trusted.pem file by concatenating the contents of iTechPoz-trusted.pem of Server1 and iTechPoz-trusted.pem of Server2, as follows:.

    - **Windows**

      ```
      type iTechPoz-trusted.pem (of Server1) >> iTechPoz-trusted.pem (of ServerN)
      ```

    - **UNIX**

      ```
      cat iTechPoz-trusted.pem (of Server1) >> iTechPoz-trusted.pem (of ServerN)
      ```

      Note: The iTechPoz-trusted.pem file is found in %DXHOME%\config\ssld directory.

12. Repeat the concatenation of certificate files of Server1 with all the other servers in the failover setup.
13. Concatenate the resulting file with certificate files from all other servers in the failover setup.
14. Copy the new iTechPoz-trusted.pem to Server1 and other servers in the failover setup to overwrite the existing files.
15. Enter the following commands to stop and start all services:

    - **Windows**

      ```
      dxserver stop all

      ssld stop

      ssld start

      dxserver start all
      ```

    - **UNIX**

      ```
      su - dsa -c "dxserver stop all"

      su - dsa -c "ssld stop"

      su - dsa -c "ssld start"
      ```

```
su - dsa -c "dxserver start all"
```

Examples for Configuring Knowledge Files

The following examples contain sample code for configuring the following files for data store failover:

- Data knowledge file (iTechPoz-Server1.dxc) to add the host name of server
- Router knowledge file (iTechPoz-Server1-Router.dxc) to add the host information in the router
- Group knowledge file (iTechPoz.dxg) to achieve group knowledge that all (Directory System Agent) DSAs in the domain can access

Example: Configuring Server1 Router Knowledge File

```
#
# iTechPozRouter - iTechnology rePOZitory
#
set dsa "iTechPoz-Server1-Router" =
{
prefix       = <cn iTechPozRouter>
dsa-name     = <cn iTechPozRouter><cn PozDsaServer1>
dsa-password = "{MD5}rMx+fEvmsStrQzYrrrhjjQ=="
…
};
```

Example: Configuring Server1 Knowledge File

```
#
# iTechPoz - iTechnology rePOZitory
#
set dsa "iTechPoz-Server1" =
{
prefix       = <cn iTechPoz>
dsa-name     = <cn iTechPoz><cn PozDsaServer1>
dsa-password = "{MD5}rMx+fEvmsStrQzYrrrhjjQ=="
address      = tcp "Server1" port 509, tcp localhost port 509
…
dsa-flag = multi-write
link-flags   = ssl-encryption-remote
};
```

Example: Configuring Server2 Router Knowledge File

```
#
# iTechPozRouter - iTechnology rePOZitory
#
set dsa "iTechPoz-Server2-Router" =
{
prefix       = <cn iTechPozRouter>
dsa-name     = <cn iTechPozRouter><cn PozDsaServer2>
dsa-password = "{MD5}rMx+fEvmsStrQzYrrrhjjQ=="
address      = tcp localhost port 1684
```

```
…
};
```

Example: Configuring Server2 Knowledge File

```
#
# iTechPoz - iTechnology rePOZitory
#
set dsa "iTechPoz-Server2" =
{
prefix       = <cn iTechPoz>
dsa-name     = <cn iTechPoz><cn PozDsaServer2>
dsa-password = "{MD5}rMx+fEvmsStrQzYrrrhjjQ=="
address      = tcp "Server2" port 509, tcp localhost port 509
…
dsa-flag = multi-write
link-flags   = ssl-encryption-remote
};
```

Example: Configuring Group Knowledge File

```
# iTechPoz - iTechnology rePOZitory
# Source the knowledge files of the iTechPozRouter and iTechPoz DSAs.
source "iTechPoz-Server1-Router.dxc";
source "iTechPoz-Server2-Router.dxc";
source "iTechPoz-Server1.dxc";
source "iTechPoz-Server2.dxc"
```

How to Use the Safex Utility to Import and Export CA EEM Data

You can use the Safex utility to import and export CA EEM data. The Safex utility lets you export and import the entire data store or a selected data set, and extract an exported data file to your local computer for editing.

The disadvantages of using the Safex utility include the requirements for data file editing and initial setup (certificate files, password), and that scripting is more difficult due to the editing of the data file.

You can use the Safex utility to migrate a reference CA EEM application to a target CA EEM server for disaster recovery purposes, and to automate the migration.

The following describes the process for migrating CA EEM data:

1. Create the export Safex XML file (see page 184)
2. Change the CA EEM certificate password in eiam.xml (see page 185)
3. Export the existing application from the reference CA EEM instance (see page 186)
4. Copy the exported application file to the target CA WCC server (see page 187)
5. Deregister the existing application in the target CA EEM instance (see page 188)
6. Register the exported application on the target CA EEM instance (see page 188)
7. Copy the certificate file to its secondary location on the target CA WCC server (see page 189)

Notes:

- All Safex commands are run from the directory that contains the Safex utility. On a CA WCC server, this is located as follows:

  Windows:

  WCC_ installation_root/_uninst/safex/win

  UNIX:

  *WCC_ installation_root*_uninst\safex\platform

  where *platform* is either Linux, SunOS, HP-UX, or AIX.

- The safex command includes the following parameters:

  `Safex –h machine –u EiamAdmin –p password –f file.xml`

  machine

Specifies the name of the new target CA EEM server.

password

Specifies the EiamAdmin password for the new target CA EEM server.

file

Specifies the name of the XML file that provides the data for executing the safex command.

Create the Export Safex XML File

Before you can export the CA EEM data, you must create a Safex XML file on the reference CA WCC server.

To create the export Safex XML file

1. Navigate to the Safex directory.

   Windows:

   WCC_installation_root/_uninst/safex/win

   UNIX:

   *WCC_installation_root*_uninst\safex*platform*

   where *platform* is either Linux, SunOS, HP-UX, or AIX.

2. Create an XML file similar to the following:

   ```
   :<Safex>

   <Attach label="label_name/><Export file="file_name" globafolders="y"
   globalgroups="y" globausers="n" folders="y" usergroups="y" users="y"
   calendars="y" policies="y" appobjects="y" />

   </Safex>
   ```

   Note: If you do not want to export an object in the file, you must change the y to n. For example, if you are using Active Directory or an external LDAP with CA EEM, you would enter "n" for the global entries.

3. Save the file with the name ExportWCCApp.xml.

Change the CA EEM Certificate Password in eiam.xml

Registering an application with CA EEM creates a security certificate on the CA WCC server. This certificate is created with a password that is included in the CA EEM application definition that you are exporting and is stored in the XML file.

Before registering the CA EEM certificate password, we recommend that you change the default password to a value that is specific for your enterprise and use the same password each time you export or import the CA WCC CA EEM application.

Note: If you store the exported XML file on your target CA WCC server for backup purposes, you should modify the clear text password so that the password is not available.

From the target CA WCC server, you will need to create a munged password using the Safex utility, then use the munged password value to edit the appropriate eiam.xml files in the CA WCC configuration directories.

To change the CA EEM certificate password

1. Enter the following command in the safex directory:

   `Safex –munge password`

   password

   Specifies the password value you plan to use in the CA EEM application.

   Note: The command returns a machine-specific munged password value. You cannot use this value on any other CA WCC instances, even if the password is the same.

2. Navigate to the working directory in the Configuration folder, as follows:

   Windows:

   WCC_installation_root\Configuration\config\working

   UNIX:

   WCC_installation_root/Configuration/config/working

3. Open eiam.xml and locate the XML tag that identifies the CA EEM server.

   It should look like this:

   `<authserver host="machine">`

4. Verify that the value of the authserver host parameter (*machine*) is the correct target CA EEM server name.

5. Locate the appcertpwd property value entry and enter the munged value of the password that you generated in Step 1.

6. Save the eiam.xml file.

7. Open ctrl.dat in a text editor to identify the active timestamp directory. Then, navigate to that directory, and open the eiam.xml file.
8. Locate the XML tag that identifies the CA EEM server, and verify that the value of the authserver host parameter (*machine*) is the correct target CA EEM server name.
9. Locate the appcertpwd property value entry and enter the munged value of the password that you generated in Step 1.
10. Save the eiam.xml file.

Export the Existing Application from the Reference CA EEM Instance

After changing the CA EEM Certificate password, you must export the existing application from the reference CA EEM server.

To export the existing application, open a command prompt, navigate to the safex directory, and enter the following safex command:

Windows:

```
safex –h EEM_servername –u EiamAdmin –p EiamAdmin_password –f ExportWCCApp.xml
```

UNIX:

```
safex –h EEM_servername –u EiamAdmin –p EiamAdmin_password –f ExportWCCApp.xml
```

EEM_servername

Specifies the name of the CA EEM server.

EiamAdmin_password

Specifies the password for the EiamAdmin.

The WCC0002-import.xml file, containing the WCC0002 application definition from the reference CA EEM server, is created.

Note: This procedurestep assumes that you already created the ExportWCCApp.xml file.

More information:

Create the Export Safex XML File (see page 184)

Copy the Exported Application File to the Target CA WCC Server

After exporting the existing application from the reference CA EEM server, you must copy the WCC0002-import.xml file to a location on the CA WCC server.

To copy the WCC0002-import.xml file, create a directory on the CA WCC server that accesses the target CA EEM instance and then copy the file to that location.

Note: This location will be used for subsequent migrations.

Change the Password in the XML File

We recommend that you change the value of the password in the exported application to the value you have identified for the target server.

Important! For CA EEM to work correctly, the munged value of this password must match what is stored in the active eiam.xml files in CA WCC Configuration.

To change the value of the password, open the WCC0002-import.xml file, and enter the following:

```
<Safex>
        <Attach/>
        <Register certfile="certfile.p12" password="password">
                <ApplicationInstance name="CA Workload Control Center"
    label="WCC0002">
```

password

Specifies the munged value of the password.

Deregister the Existing Application in the Target CA EEM Instance

You must deregister the existing application in the target CA EEM server.

To deregister the current application on the target CA EEM server, open a command prompt, navigate to the safex directory, and enter the following command:

Windows:

```
safex –h EEM_servername –u EiamAdmin –p EiamAdmin_password –f
..\UWCCUnRegister.xml
```

UNIX:

```
safex –h EEM_servername –u EiamAdmin –p EiamAdmin_password –f
../UWCCUnRegister.xml
```

EEM_servername

Specifies the name of the CA EEM server.

EiamAdmin_password

Specifies the password for the CA EEM administrator.

Register the Exported Application on the Target CA EEM Instance

You must register the imported application with the target CA EEM server.

To register the imported application, open a command prompt, navigate to the safex directory, and enter the following command:

Windows:

```
safex -h EEM_servername -u EiamAdmin -p admin -f file_path\WCC0002-import.xml
```

UNIX:

```
safex -h EEM_servername -u EiamAdmin -p admin -f file_path/WCC0002-import.xml
```

EEM_servername

Specifies the name of the CA EEM server.

file_path

Specifies the location of the WCC0002-import.xml file.

Copy the Certificate File to the Secondary Location on the Target CA WCC Server

The final step in the process requires you to copy the certificate file to the secondary location on the CA WCC server.

To copy the certificate file, do one of the following:

- **Windows:**

  Copy the cert-uwcc.p12 certificate to *WCC_instalation_root*\ConfigServer\config

- **UNIX:**

  Copy the cert-uwcc.p12 certificate to *WCC_installation_root*/ConfigServer/config

How to Use the CA Directory Commands

You can use the CA Directory commands to extract and reload the CA EEM database. This method is easy to script and requires no editing. However, it does perform a full database download and reload, including the deleted items, and it cannot be used with CA EEM 8.4 and above.

The following describes the process for using the CA Directory Commands to extract and reload the CA EEM database:

1. Run the commands on the reference server (see page 190)
2. Run the commands on the target server (see page 191)
3. Verify the CA EEM policies on the target server (see page 191)

Run the CA Directory Commands on the Reference Server

You can use the CA Directory commands on the reference server to download the CA EEM database.

To run the CA Directory commands on the reference server

1. Create two new shared folders to store database dumps and backups as follows:

   reference_server\CA\EEM_Backup (D:\CA\EEM_Backup)

   reference_server\CA\EEM_Dump (D:\CA\EEM_Dump)

2. Run the dxdumpdb command as follows:

   ```
   dxdumpdb -S itechpoz-referenceserver -p "cn=itechpoz" itechpoz >
   \\reference_server\CA\EEM_Dump\reference.ldif
   ```

3. Run the ldifsort command as follows:

   ```
   ldifsort \\reference_server\CA\EEM_Dump\reference.ldif
   \\reference_server\CA\EEM_Dump\sorted_reference.ldif
   ```

4. Run the xcopy command as follows:

   ```
   xcopy /Y/V/F "\\reference_server\CA\EEM_Dump\sorted_reference.ldif"
   \\target_server\EEM_Load
   ```

Run the CA Directory Commands on the Target Server

You can use the CA Directory Commands to load the CA EEM database on the target server.

To run the CA Directory Commands on the target server

1. Create two new shared folders in which to store database pending loads and backups as follows:

   target_server\CA\EEM_Load (C:\Program Files\CA\EEM_Load)

   target_server\CA\EEM_Backup (C:\Program Files\CA\EEM_Backup)

2. Open a command prompt and run the following commands:

   a. Create a restore capability by entering the following:

      ```
      dxbackupdb itechpoz
      ```

   b. Stop the dxserver using the following command:

      ```
      dxserver stop all
      ```

   c. Empty the database using the following command:

      ```
      dxemptydb itechpoz
      ```

   d. Enter the dxloaddb command to load the database, as follows:

      ```
      dxloaddb -S itechpoz-target_server -p "cn=itechpoz" itechpoz < sorted_reference.ldif
      ```

   e. Enter the following command to tune the database:

      ```
      dxtunedb itechpoz
      ```

   f. Enter the dxserver start command to restart the dxserver, as follows:

      ```
      dxserver start all
      ```

   The target server has been loaded with the current CA EEM database and is ready for verification.

Verify the CA EEM Policies

After you load the CA EEM database on the target server and configure it, you must verify the CA EEM policies.

To verify the CA EEM policies, open CA EEM on the target server, log in to the CA WCC application, and verify your policies.

All of the policies from the reference CA EEM server are now created on the target server.

Appendix A: CA ELM and Event Reports

Register CA ELM for CA EEM r8.4

CA ELM (CA Enterprise Log Manager) is used to generate CA EEM events and event reports. CA ELM monitors server status processes, status of agents, agent details, and connection details. You must import and register CA ELM before you can use the reports.

To import and register CA ELM

1. Open a command prompt.
2. Navigate to the iTechnology directory. By default, this is c:\Program Files\CA\SharedComponents\iTechnology on Windows or /opt/CA/SharedComponents/iTechnology on UNIX.
3. Enter the following:

   `Safex –h hostname -u eiamadmin –p eiamadmin_password -f reg.xm`

   where

   -h *hostname*

   Specifies the name of the CA EEM server.

   -p *eiamadmin_password*

   Specifies the password for the eiamadmin user.
4. Press Enter.

   CA ELM is imported and registered on the CA EEM server.

Index

U

V