

CA GREEN BOOKS

# CA Workload Automation DE

Deployment Best Practices

Version 1.0

- OVERVIEW OF CA WORKLOAD AUTOMATION DE
- BEST PRACTICES FOR DEPLOYING THE CA WORKLOAD AUTOMATION DE SOLUTION



## LEGAL NOTICE

This publication is based on current information and resource allocations as of its date of publication and is subject to change or withdrawal by CA at any time without notice. The information in this publication could include typographical errors or technical inaccuracies. CA may make modifications to any CA product, software program, method or procedure described in this publication at any time without notice.

Any reference in this publication to non-CA products and non-CA websites are provided for convenience only and shall not serve as CA's endorsement of such products or websites. Your use of such products, websites, and any information regarding such products or any materials provided with such products or at such websites shall be at your own risk.

Notwithstanding anything in this publication to the contrary, this publication shall not (i) constitute product documentation or specifications under any existing or future written license agreement or services agreement relating to any CA software product, or be subject to any warranty set forth in any such written agreement; (ii) serve to affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (iii) serve to amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this publication remain at CA's sole discretion.

The information in this publication is based upon CA's experiences with the referenced software products in a variety of development and customer environments. Past performance of the software products in such development and customer environments is not indicative of the future performance of such software products in identical, similar or different environments. CA does not warrant that the software products will operate as specifically set forth in this publication. CA will support only the referenced products in accordance with (i) the documentation and specifications provided with the referenced product, and (ii) CA's then-current maintenance and support policy for the referenced product.

Certain information in this publication may outline CA's general product direction. All information in this publication is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "AS IS" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill or lost data, even if CA is expressly advised of the possibility of such damages.

### **COPYRIGHT LICENSE AND NOTICE:**

This publication may contain sample application programming code and/or language which illustrate programming techniques on various operating systems. Notwithstanding anything to the contrary contained in this publication, such sample code does not constitute licensed products or software under any CA license or services agreement. You may copy, modify and use this sample code for the purposes of performing the installation methods and routines described in this document. These samples have not been tested. CA does not make, and you may not rely on, any promise, express or implied, of reliability, serviceability or function of the sample code.

Copyright © 2015 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. Microsoft product screen shots reprinted with permission from Microsoft Corporation.

### **TITLE AND PUBLICATION DATE:**

*CA Workload Automation DE Deployment Best Practices Green Book Version 1.0*  
Publication Date: March, 2015

# ACKNOWLEDGEMENTS

**Principal Author**

Nitin Pande

## CA TECHNOLOGIES PRODUCT REFERENCES

This document references the following CA Technologies products:

- CA Workload Automation Agents

## FEEDBACK

Please email us at [greenbooks@ca.com](mailto:greenbooks@ca.com) to share your feedback on this publication. Please include the title of this publication in the subject of your email response. For technical assistance with a CA Technologies product, please contact CA Support at <https://support.ca.com>. For assistance with support specific to Japanese operating systems, please contact CA Technologies at <http://www.casupport.jp>.

# Contents

<b>LEGAL NOTICE</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iii</b>
<b>CA TECHNOLOGIES PRODUCT REFERENCES</b>	<b>iv</b>
<b>FEEDBACK</b>	<b>iv</b>
<b>Contents</b>	<b>5</b>
<b>Chapter 1: Introduction</b>	<b>7</b>
<b>Chapter 2: Deployment Architecture for CA Workload Automation DE</b>	<b>9</b>
CA Workload Automation DE Server .....	10
Planning Considerations .....	10
Artifacts .....	12
CA WA Desktop Client .....	13
CA Workload Automation Agents .....	13
Command Line Interface (CLI) .....	13
CA WA Web Services .....	14
<b>Chapter 3: CA WA Desktop Client and APIs</b>	<b>15</b>
CA WA Desktop Client Installation .....	15
Command Line Interface (CLI) .....	17
Stand-alone CLI .....	17
CA WA Web Services .....	18
How the Web Services Work .....	19
Program your Software Applications to Use Web Services .....	19
Web Services Functions .....	20
<b>Chapter 4: Server Installation</b>	<b>21</b>
Installing on virtual platform .....	21
Network Interface .....	22
CA Workload Automation Ports .....	22
Email Setup .....	22
SNMP Messages .....	23
Log Files .....	24
Trace Log .....	24
Error Log .....	24
Audit Log .....	24
Buffer Log .....	24
Logging Profiles .....	25

<b>Chapter 5: CA WA High Availability</b>	<b>26</b>
CA WA High Availability Configuration .....	26
How CA WA High Availability Works.....	27
Stage 1 .....	27
Stage 2 .....	28
Failover .....	29
Failback .....	29
<b>Chapter 6: Scheduling</b>	<b>31</b>
Applications .....	31
Jobs .....	31
Events .....	32
JavaScript.....	34
Resources .....	35
Global Variables.....	36
Calendars.....	36
Application LifeCycle.....	37
Job LifeCycle .....	38
<b>Chapter 7: Administration</b>	<b>40</b>
Housekeeping Application .....	40
PURGELOG .....	41
Workload Maintenance .....	42
Monitoring Considerations.....	43
Minimum Free Memory.....	44
Agent Heartbeat .....	44
<b>Chapter 8: Database Requirements</b>	<b>47</b>
Database Setup.....	48
Database Maintenance.....	48
PURGECOMPLETEDJOBS .....	48
MOVEHISTORYDATA .....	49
DELETETATUSMESSAGES.....	49
Increase Database Connections .....	50
Idle Connections .....	51
Database Tuning .....	51
Oracle Database Considerations .....	51
SQL Server Considerations .....	52
<b>Chapter 9: LDAP</b>	<b>53</b>
Multiple Authentication Systems .....	55
<b>Chapter 10: Additional Considerations</b>	<b>56</b>
Agent Groups.....	57
Implementation Checklist.....	57

# Chapter 1: Introduction

Welcome to the CA Workload Automation DE (dSeries Edition) Green Book. This series describes the best practices for deploying your Workload Automation DE solution. The CA solution presents an enterprise and business-wide view of the business process as it is executing, offers tailored views based on the role of the solution user, and provides granular role-based security. The solution also matches work to do with the resources available to do the work according to the business process policies that define the service levels.

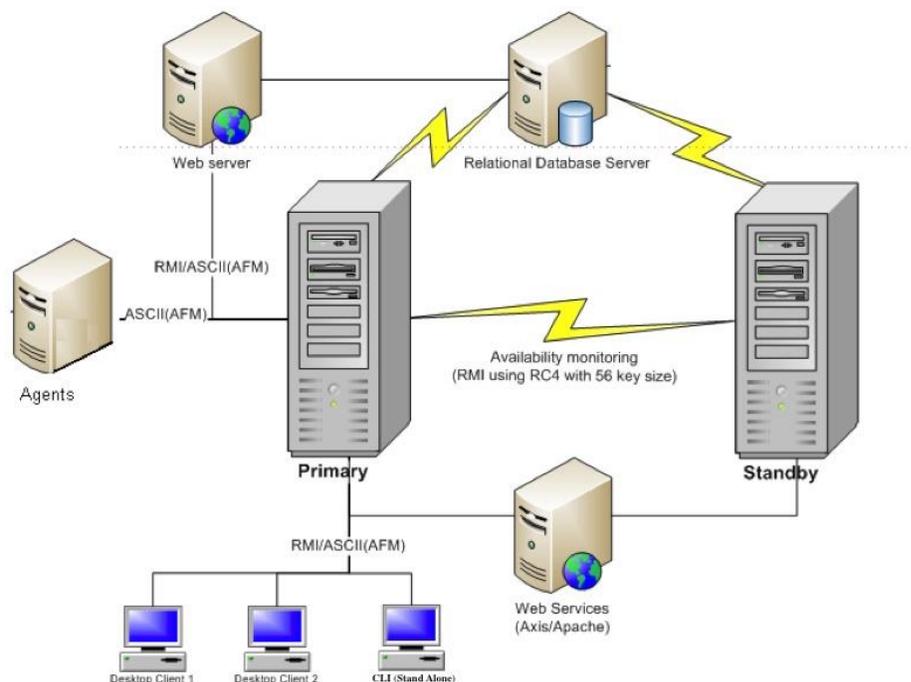
The information contained in this Green Book is designed for the technical staff charged with deploying and administering CA Workload Automation DE, as well as staff responsible for a line of business, production control, and security.



# Chapter 2: Deployment Architecture for CA Workload Automation DE

CA Workload Automation DE provides distributed job scheduling and workload management across the enterprise. It is a simple, flexible, and powerful solution for enterprise application integration (EAI) and systems operations. The CA Workload Automation DE system consists of the following required components:

- CA Workload Automation DE server
- CA WA Desktop Client
- CA Workload Automation Agent



The suite also includes the following optional components.

- CA Workload Automaton Web Services
- CA Workload Automaton CLI
- CA Workload Automaton Web Client

## CA Workload Automation DE Server

---

The server also named manager or scheduler is the main repository of all the workload. The server is a Java application and is bundled with JRE. The server can be installed in a stand-alone or high-availability mode on Linux, UNIX, or Windows operating system.

The DE server is a Java-based multi-threaded application. It maximizes the efficiency of CPU. As per the Java requirement, the server has pre-allocated heap size of 1 GB. However, the server only acquires what it needs.

UNIX and Linux:

The maximum heap size is set using the `MAX_HEAP_SIZE` parameter in the `<install_directory>/bin/startServer` script:

```
MAX_HEAP_SIZE=1024
```

Windows:

All the JVM settings include the heap size set in the `<install_directory>\conf\windows.service` file:

```
jvmproperty_3=-Xmx4096M
```

**Important!** When you plan to install CA Workload Automation DE, the 64-bit version is always recommended. However, 64-bit version can only be installed on a 64-bit operating system. One of the advantages of the 64-bit version is that it allows larger (>2 GB) heap size.

In a production setup, the heap size must be increased to 2 GB or more. If you installed the 64-bit server, increase the heap size to 4 GB or 4096 MB. The server must have a defined heap available and free always. Any changes to the server configuration require a server restart.

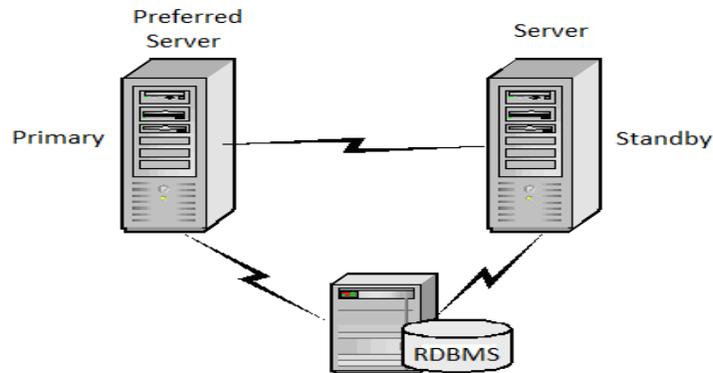
Any user can install and start the server. It is not necessary to install the server as a root user or as an administrator. It is also not required or recommended to update the bundled JRE at any point. Normally, the Service Packs update the JRE if necessary. The server JRE must not be used by any other external application or process.

### Planning Considerations

The server requires a relational database for message processing, CA WA High Availability, and storing server configuration files, resource definition files, and historical reporting data. For the latest database support matrix, refer to this link:

<https://support.ca.com/irj/portal/anonymous/phpsupcontent?contentID=B33BBAF8-3EA1-4D42-834E-788F93B98B5D>

**Important!** CA Workload Automation DE supports several database types. The database connection must be reliable and constant. Network latency of below 5 milliseconds provides optimal performance.



The database schema is logically divided into runtime, history, definition, and staging tables.

### Runtime Tables

The runtime tables store queue and persistent data. These are highly transactional tables. The server constantly writes and updates these tables. For example, ESP\_RTWOB table contains information of all the currently running jobs.

### History Tables

The history tables stores information of completed jobs and application dates. These tables tend to grow over time and require maintenance. The ESP\_APPLCIATION and ESP\_GENERIC\_JOB tables store historical data of completed application generations and jobs. New built-in commands available in R11.3 allows easy maintenance of these tables.

**Note:** For more information about the usage of built-in commands, see [Database Requirements](#).

### Definition Tables

The Definition tables store the information about the entire job, application, events, and all other artifacts. The artifacts are defined in XML format. These XML files are stored as CLOB data types in the database. For example, the ESP\_EVENT\_DF table stores all the events.

## Staging Tables

The staging tables, introduced in r11.3, store historical data that are moved from active (ESP\_) tables. These tables are not accessed or needed for normal running of the workload. All these tables start with H\_. Each of these tables has a corresponding ESP\_ table. For example, H\_APPLICATION table has the same structure as ESP\_APPLICATION. When the maintenance command is issued, the data from ESP\_<table> is moved to structurally identical H\_<table>. Not all ESP\_ or active tables have a matching H table.

The list of H tables is as follows:

- H\_APPLICATION
- H\_AS400\_JOB
- H\_BDC\_JOB
- H\_BWIP\_JOB
- H\_BWIC\_JOB
- H\_FILEMONITOR\_JOB
- H\_GENERIC\_JOB
- H\_PEOPLESOFT\_JOB
- H\_SAP\_JOB
- H\_SPDA\_JOB
- H\_SPPM\_JOB
- H\_VIRTUAL\_JOB

## Artifacts

The Artifacts, entities within CA Workload Automation DE, lets users run simple to complex workloads. The artifacts are defined in XML format. They are stored in CLOB data format in the database.

The list of artifacts is as follows:

- Agent
- Agent Groups
- Alert
- Application
- Calendar
- Event
- Forecast
- Global Variables

- Group
- JavaScript
- Reports
- Resources
- Sundry
- Users

You can use import-export utility to export and import all the artifacts from one instance of DE to another.

## CA WA Desktop Client

---

CA WA Desktop Client is a graphical interface with clean and uncluttered interface.

The interface lets user drag -and-drop workload definitions, manage calendars, and monitor and control batch workloads. It also includes administrator's tools for setting up, monitoring, and diagnosing problems with CA Workload Automation DE.

## CA Workload Automation Agents

---

The CA Workload Automation Agent harnesses the power and performance of Java. Java has become ubiquitous with systems; it is integrated into all-major operating systems. Another major benefit of Java is its security. The language and the platform were designed from the ground up with security in mind.

With powerful features, the agent keeps minimal overhead on the hosts. The agent installers are bundled with JRE on most of the platforms. Only i5 and z/Linux operating system require IBM JRE installation before the agent installation.

The server is installed with a default agent. For example, if you install the server on Red Hat Linux, a CA WA Agent for Linux is automatically installed on the same computer in the server installation directory:

```
/opt/CA/WorkloadAutomation_R11.3/ESPSystemAgent
```

## Command Line Interface (CLI)

---

The Command Line Interface (CLI) lets you issue commands against the server and agents. For example, you can issue the STOP command to stop the server. You can also issue common scheduling, operations, and programming commands. It is installed in the <server\_install\_directory>/bin directory. It can also be easily accessed using CA WA Desktop Client.

## CA WA Web Services

---

CA WA Web Services lets you integrate the server in a Service Oriented Architecture (SOA) using Simple Object Access Protocol (SOAP) over HTTP or HTTPS. By using web services, you do not need proprietary software to integrate the server with other services in a business process.

# Chapter 3: CA WA Desktop Client and APIs

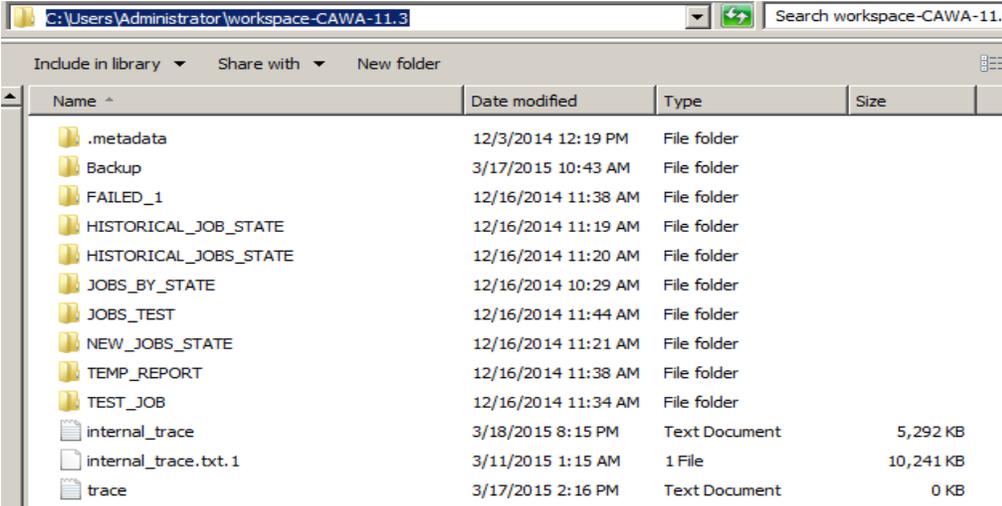
The CA WA Desktop Client lets users, operators, and administrators interact with the scheduler. Similar to the server and agents, the Client is also a Java application. It is packaged with JRE. The Client requires its own installation directory.

## CA WA Desktop Client Installation

You can install CA WA Desktop Client using an interactive wizard that lets you change and review your settings before starting the installation process. To install CA WA Desktop Client, run the installation program provided on the product DVD. The installation program installs all of the files required to run the interface. Administrator or user with full admin privileges must execute the client installer.

**Important!** The default installation directory for CA WA Desktop Client is C:\Program Files for 32-bit operating systems and C:\Program Files (x86) for 64-bit operating systems. We recommend that you use the default location for the installation. If C: is not your primary drive, change the drive letter to match the Windows operating system primary volume.

When a user runs the client, a workspace directory is created in the user's home directory. The workspace directory includes local settings information, client logs, and report projects that are created on the client.



Name	Date modified	Type	Size
.metadata	12/3/2014 12:19 PM	File folder	
Backup	3/17/2015 10:43 AM	File folder	
FAILED_1	12/16/2014 11:38 AM	File folder	
HISTORICAL_JOB_STATE	12/16/2014 11:19 AM	File folder	
HISTORICAL_JOBS_STATE	12/16/2014 11:20 AM	File folder	
JOBS_BY_STATE	12/16/2014 10:29 AM	File folder	
JOBS_TEST	12/16/2014 11:44 AM	File folder	
NEW_JOBS_STATE	12/16/2014 11:21 AM	File folder	
TEMP_REPORT	12/16/2014 11:38 AM	File folder	
TEST_JOB	12/16/2014 11:34 AM	File folder	
internal_trace	3/18/2015 8:15 PM	Text Document	5,292 KB
internal_trace.txt.1	3/11/2015 1:15 AM	1 File	10,241 KB
trace	3/17/2015 2:16 PM	Text Document	0 KB

The Desktop Client has seven components named perspectives each providing major functionalities:

- Define
- Monitor
- Services
- Admin
- CLI
- Report
- SAP Tools



The Define perspective lets users create, edit or update an Application, jobs, and events. The Monitor perspective lets users monitor application and job status. The Services perspective lets users create or update the following artifacts:

- Alerts
- Calendars
- Events
- Forecasts
- Global Variables
- JavaScripts
- Reports
- Resources

The Reports perspective lets users create special reports using packaged Business Intelligence and Reporting Tools (BIRT). The report workspace stores all the locally created reports in the Desktop Client workspace. The reports can be uploaded and made available to other users.

The SAP Tools perspective lets users monitor and control an SAP system. SAP Tools is a window into SAP systems that lets you do the following:

- List and manually trigger SAP events
- Work with intercepted jobs
- Monitor jobs CA Workload Automation DE did not submit

- Determine current resource usage on the SAP system

The Admin perspective lets administrators perform tasks associated with the server and agents. The administrators can add users, define agents in the Topology, and configure various other parameters.

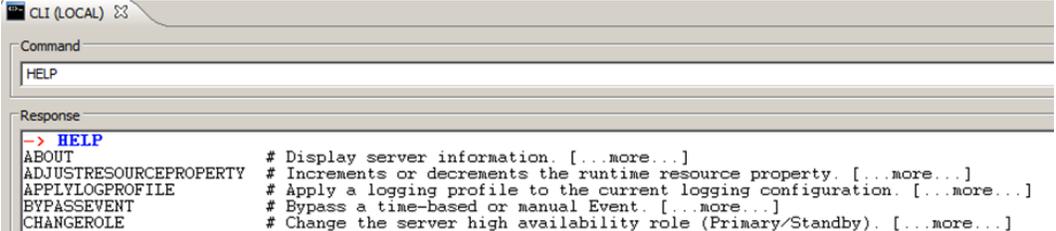
## Command Line Interface (CLI)

The server is installed with a stand-alone CLI in the bin directory. The CLI is also a perspective of CA WA Desktop Client. It provides the same functionality as the stand-alone CLI utility. The cli shell script on UNIX or cli.bat batch script on Windows invokes this utility.

To use the CLI, you must be connected to the server. The commands that you can issue depend on your security permissions. CLI commands can be used to work with the following:

- Server
- Agents
- Server log files
- Applications and jobs
- Events
- Global variables and variable dependencies
- Resources

You can view the list of available CLI commands by using the HELP command.



```

CLI (LOCAL)
Command
HELP
Response
-> HELP
ABOUT # Display server information. [...more...]
ADJUSTRESOURCEPROPERTY # Increments or decrements the runtime resource property. [...more...]
APPLYLOGPROFILE # Apply a logging profile to the current logging configuration. [...more...]
BYPASSEVENT # Bypass a time-based or manual Event. [...more...]
CHANGEROLE # Change the server high availability role (Primary/Standby). [...more...]

```

### Stand-alone CLI

The stand-alone CLI utility can also be installed on other UNIX or Windows computers. The stand-alone CLI installer is shipped in the CA Workload Automation DE DVD.

The syntax for connecting through CLI is as follows:

```
cli | cli.bat <DE server IP or hostname> <DE port> <DE user> <user password>
command
```

### Example: CLI Connection Session

```
[root@de-server bin]# ./cli de-server 7500 admin admin_pass
Connected
admin@de-server:7500(PRIMARY)> about
Copyright (c) 2014 CA. All rights reserved.
http://www.ca.com/us/products/product.aspx?id=7833

CA Workload Automation Server Id: DSERIES_7500
CA Workload Automation Server Version: 11.3.3.0
Build: 1342
Build Date: 20141203
High Availability is NOT Enabled
Connected clients:
  ADMIN@xxx.example.com:60297 (10.133.10.10:60297) connection confirmed
  ADMIN@localhost:50792 (127.0.0.1:50792)
```

### Example: Use CA Workload Automation DE Commands as part of the CLI Switches or a Script

```
[root@de-server bin]# ./cli de-server 7500 admin admin_pass about
Copyright (c) 2014 CA. All rights reserved.
http://www.ca.com/us/products/product.aspx?id=7833

CA Workload Automation Server Id: DSERIES_7500
CA Workload Automation Server Version: 11.3.3.0
...
```

**Note:** The following variables can set in the cli script or can be exported before running cli:

```
PREFERRED_SRV=de-server.example.com
PREFERRED_SRV_PORT=7500
STANDBY_SRV=de-standby.example.com
STANDBY_SRV_PORT=7500
```

You may also set these variables in the user profile.

### Examples: Use Variables with the CLI Command

UNIX/Linux:

```
cli $PREFERRED_SRV $PREFERRED_SRV_PORT admin pass command
```

Windows:

```
cli.bat %PREFERRED_SRV% %PREFERRED_SRV_PORT% admin pass "command"
```

## CA WA Web Services

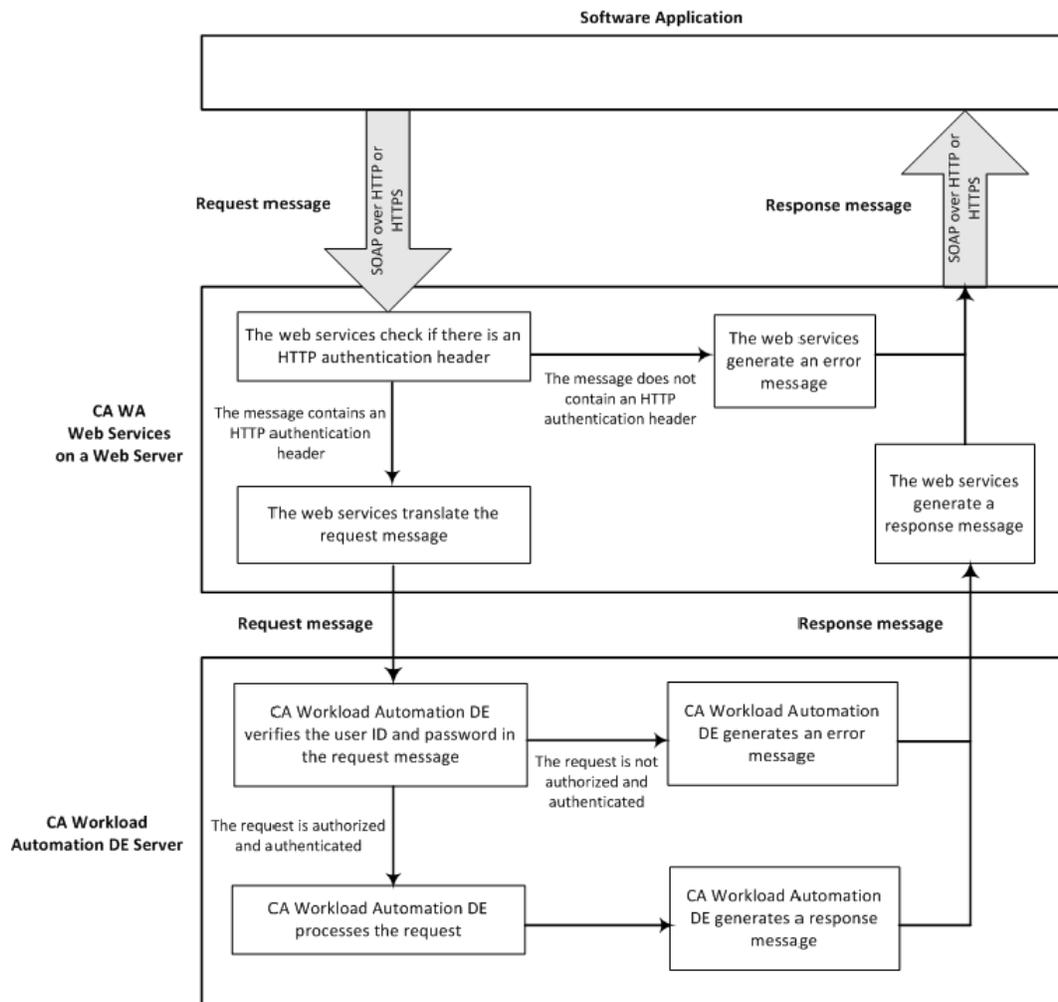
---

You can program any software application that is configured to work with web services to create, update, invoke, monitor, and control CA Workload Automation DE workload. For example, you can install the CA WA Web Services and program a customized internal web interface to integrate with CA Workload Automation DE.

### How the Web Services Work

An external software application can create, update, invoke, monitor, and control CA Workload Automation DE workload by sending a message to the web services that are installed on a web server. The web server then sends the message to the CA Workload Automation DE server, which processes the request. The messages are sent using Simple Object Access Protocol (SOAP) over HTTP or HTTPS.

The following diagram shows how the components communicate:



### Program your Software Applications to Use Web Services

You can program your web services applications to call the functions that access the CA WA Web Services. Calling the CA Workload Automation functions depends on your programming languages and tools. For more information about programming your software applications to work with web services, see the documentation for your tools.

These functions and the expected responses are described in the Web Services Description Language (WSDL) file that is provided with the web services. The functions generate SOAP messages that are sent over HTTP or HTTPS to the web services.

### Web Services Functions

You can use web services to invoke various types of functions against events, applications, jobs, resources, global variables, and artifacts. Some of the available operations are as follows:

- createVariable
- unbypassJob
- runApplicationRead
- getServerTimeZone
- runApplicationCreate
- listEventSchedule
- dropResourceDependency
- setResourceMaxAvailability
- suspendEvent
- triggerReplaceEvent
- listActiveApplicationsForEvent
- releaseApplication
- holdEvent
- cancelActiveJob
- deleteVariableContext

For the complete list of web services functions, log in to the Web Services using a browser using the following URL:

[http://<webservice\\_server>:8080/axis2/services/listServices](http://<webservice_server>:8080/axis2/services/listServices)

**Note:** You are required to enter the user credentials to log in. The default password for user 'admin' is changeit.

# Chapter 4: Server Installation

Physical deployment is an important factor for performance. However, a common mistake in performance and scalability tuning is to initially focus on scaling up or scaling out the hardware. You can achieve high performance by minimizing resource usage. You must consider adding hardware only after you limit CPU use, network I/O, disk I/O, and memory use.

When installing the server, ensure that your computer meets the system requirements. The DE server can be installed on Linux, UNIX, and Windows operating system. The system must have the following recommended or better configuration for optimal performance:

Installation Recommendations	
CA Workload Automation DE r11.3	Latest 64-bit version on 64-bit operating system
Disk space	40 GB free
Memory	8-12 GB
CPU	4 CPU cores, 1.5 GHz to 3 GHz each
Network	1 Gigabit or better (with less than 5 ms latency between Primary and Standby hosts)

**Note:** For more information about the system requirements, see the Release Notes in the product DVD or at the [CA Workload Automation DE DocOps platform](#).

## Installing on virtual platform

---

Virtual environments are dynamic bodies, where virtual machines (VM) may be constantly on the move between virtual hosts. Hence, the processing and memory utilization, including all physical resources, are used in much higher levels than in traditional physical setup. The overall densities are higher, in storage, resource utilization, and servers and applications.

When installing the server on a VM, ensure that you allocate specific amount of resources to the images.

- Allocate specific percentage of total memory to the VM
- Allocate some CPU cycles to increase performance
- Use SAN storage for better I/O
- Disable auto movement of images, whenever possible
- Set alarms or notifications when VM requires attention or use large amount of resources.

## Network Interface

The server installation program prompts you to specify the network interface for communication. You can enter the IP address or the host name of the local host that the server uses for communication.

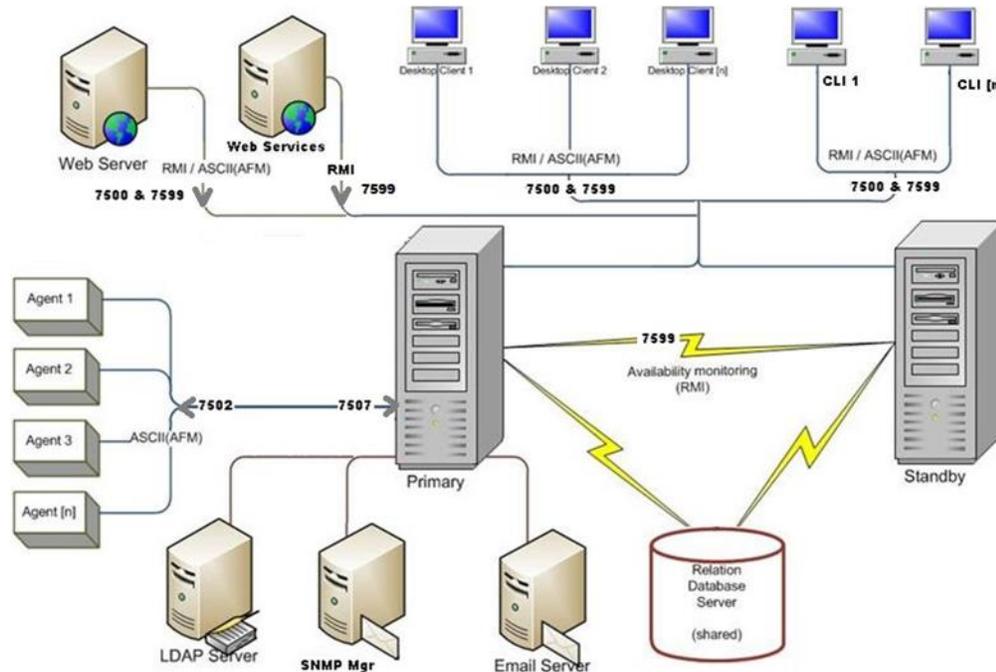
**Important!** When specifying a host name, use Fully Qualified Domain Name (FQDN). The server FQDN must be resolvable using DNS or a name resolution service. It also minimizes changes on the server if IP address modifications are required.

## CA Workload Automation Ports

By default, the CA Workload Automation DE uses these ports: 7500, 7507, 7599 and 7598.

**Important!** Use the default ports unless they are used by some other services. If there is any local or network firewall, then it must have these ports open.

For example, port 7507 is for incoming traffic; it has to be open from agent hosts. The Desktop Client uses ports 7500 and 7599 to communicate with the server; they must be open as incoming to the server from the Client host.



## Email Setup

The server can send different types of job, application, and server-level notifications. Specify the server host name or IP address with an e-mail address. The server cannot authenticate against messaging server; it has to relay the mail.

## SNMP Messages

You can set up your server to work with SNMP (Simple Network Management Protocol). The server and SNMP-enabled agents use SNMP messages to notify users of their activity such as server start-up and shutdowns, agent inactivity or shutdown, and CA WA High Availability messages. You must configure the SNMP Message Viewer on CA WA Desktop Client or at least one SNMP manager.

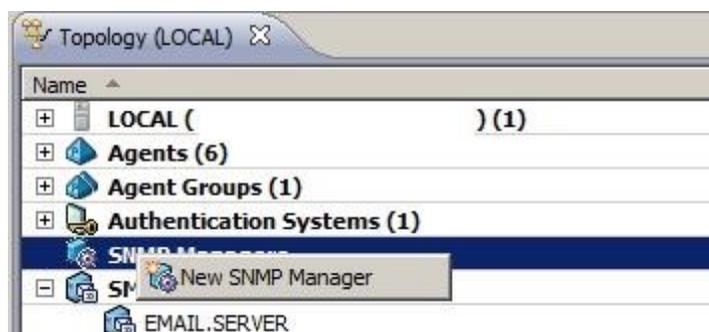
The server sends an SNMP trap for the following events to the SNMP Message Viewer or SNMP manager that you configured:

- Each time the server starts or stops
- Each time an agent starts or stops
- When the monitoring server reaches the maximum number of consecutive, unsuccessful polling attempts in a CA WA High Availability configuration. It indicates a possible problem with the server
- When the connection to the LDAP server is established or lost
- When an error in the logging component prevents the server from logging messages to the trace log or buffer log, for example, if the disk is full

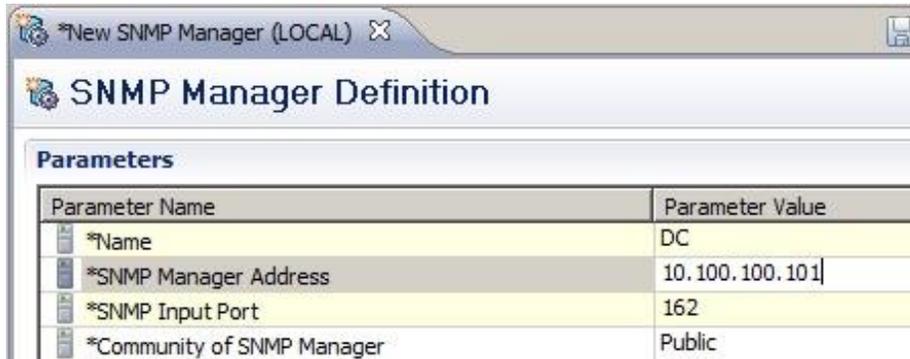
CA WA Desktop Client can be used as an SNMP Message Viewer to monitor traps sent from servers, agents, or jobs.

To add an SNMP manager,

1. In the Admin Topology, right-click the SNMP Managers node, and select New SNMP Manager from the pop-up menu.



2. Add the local address of the Desktop Client in the SNMP Manager Definition view.



## Log Files

---

The server maintains the following log files to store diagnostic and auditing information:

### Trace Log

The tracelog stores communication messages between the CA Workload Automation DE components and maintains debugging information. You can filter the messages that the server writes to the trace log. Its file name is tracelog.txt. By default, the trace log is archived (rolls over) once it reaches 10 MB in size.

### Error Log

The error log stores messages that are logged with severity ERROR or higher. Its file name is errors.txt. The error log never rolls over and is recreated when the server starts up. If there is a server shutdown or other issues, this log can provide information about errors or exceptions encountered by the server.

### Audit Log

The audit log records user input. The server creates an audit log every day. You cannot change this frequency. The audit log is encrypted. You cannot use the server to archive or clear the audit log automatically. Instead, you can create HTML and Comma Separated Values (CSV) reports by using the CLI. This log must be cleared manually.

### Buffer Log

The buffer log stores all messages between the server and the database regardless of category or severity. Its file name is buffer.txt.

## Logging Profiles

A logging profile is a set of logger identifiers with associated severity thresholds. You can set the severity thresholds of all the logger identifiers at once by creating and applying a logging profile instead of setting the severity threshold of each logger identifier individually.

The server provides the following predefined logging profiles:

- DEFAULT
- DEFAULT PLUS RDBMS
- ESSENTIAL
- EVERYTHING

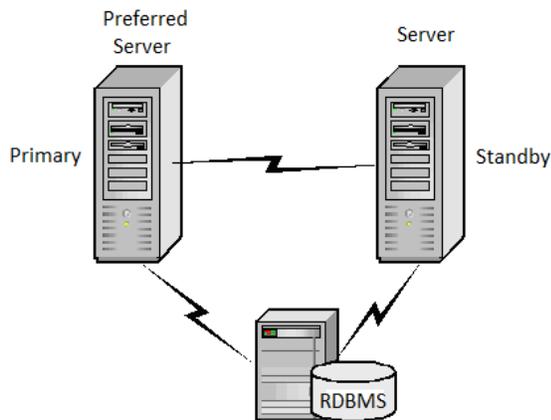
**Note:** In r11.3, the logging profile can be changed without requiring a server restart. For a stable production environment, you can set the profile to ESSENTIAL. This setting captures the basic server information. To capture additional information for debugging or monitoring an issue, use the DEFAULT to EVERYTHING profile.

# Chapter 5: CA WA High Availability

CA WA High Availability is the CA Workload Automation DE failure detection and recovery process. A CA WA High Availability configuration requires a Primary server and a Standby server. When both servers are running, the server that is active controls the workload and interacts with agents defined in the Topology to run jobs. The other server monitors the active server. When the active server fails, workload processing is switched from the active server to the monitoring server (failover).

## CA WA High Availability Configuration

A typical CA WA High Availability configuration consists of two servers installed on separate computers. Both servers use the same relational database management system (RDBMS). The Primary is the preferred server, which becomes active when both servers are started but the active and monitoring roles are not yet assigned. The preferred server also becomes active in a failback condition when automatic failback is enabled and both servers are available.



When configuring CA WA High Availability, ensure that your setup meets the following requirements:

- Install the Standby on a different computer than the Primary. The system requirements for the Primary and Standby computers are the same.

**Note:** For more information about the system requirements, see the Release Notes. For testing, you can install the Primary and Standby on the same computer, but you must use different ports.

- The Primary and Standby must run on the same operating system type and version.

- The system clocks on the two computers must be within 500 milliseconds of each other. The systems synchronize when you connect them to a time server. Use Network Time Protocol (NTP) to synchronize clocks.
- The Primary and Standby must be in the same time zone. If you install the Primary and Standby in different time zones, in a failover situation, the server might run workload twice or skip scheduled jobs.
- Keep both servers within same subnet to prevent any failover latency. In a virtual environment, VLAN may be used.
- You can install only one pair of Primary and Standby servers per CA WA High Availability configuration.

When installing the servers on virtual machines (VM), ensure that each VM is on a separate physical hardware or frame. We recommend that you disable automatic image movement as this can potentially lead to both Primary and Standby server VMs on the same physical instance.

## How CA WA High Availability Works

---

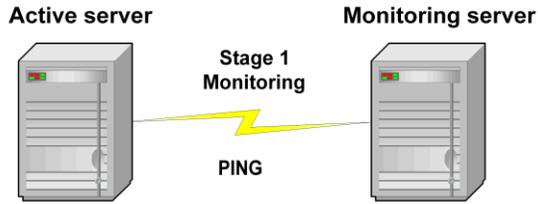
During normal processing, the active CA Workload Automation DE server runs the workload. The other or standby server monitors the active server and the database to check if failure has occurred on the active server. The servers share the same database, so no data replication or extra processing takes place. A dual-stage or two-stage failure detection process is used to determine whether failure has occurred. Failure can occur if the active server goes down or if it terminates. Network connection failure between Primary and Standby can also result in failure.

CA WA High Availability works as follows:

- Stage 1
- Stage 2
- Failover
- Failback

### Stage 1

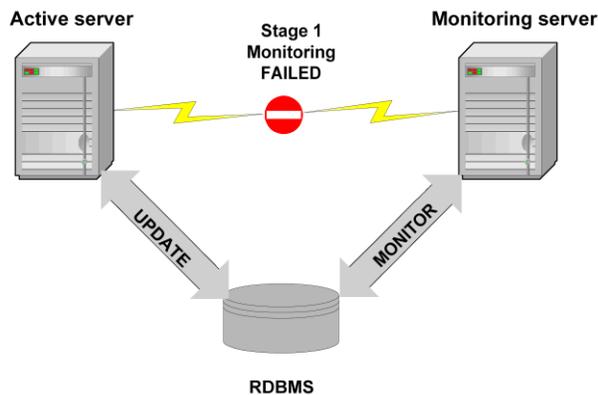
In stage 1 of the failure detection process, if both servers are running, the monitoring server periodically contacts the active server based on the configured heartbeat. The ping frequency (sec) setting determines how long the monitoring server waits for response from active server before it goes to stage 2. The default value is 60 seconds.



If the servers cannot communicate directly with each other, stage 2 monitoring takes place.

## Stage 2

In stage 2 of the failure detection process, the monitoring server monitors the relational database. As both servers require database access to operate, the server uses the database as the final arbiter to determine whether the active server has failed.



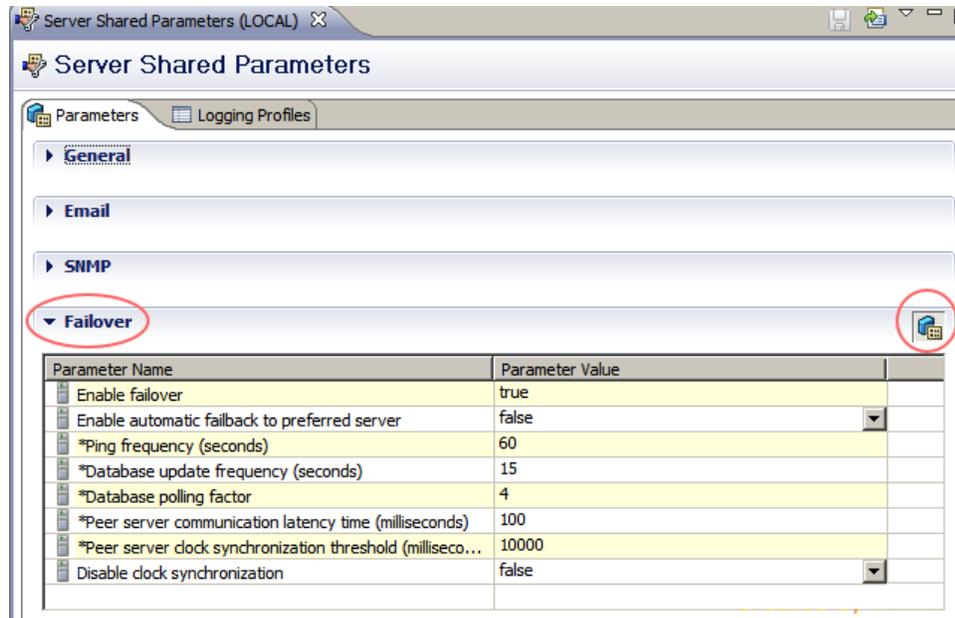
During stage 2, the active server begins to periodically update a table in the database. The monitoring server monitors (polls) this table to verify if the active server is still active. Before each database poll, the monitoring server attempts to contact the active server (stage 1 monitoring).

The database update frequency (sec) setting determines how often, in seconds, the active server updates a database table and the monitoring server polls during stage 2 failure detection.

The database polling factor setting determines the polling interval factor. This value is required to compute the polling interval, which is the frequency in which the monitoring server polls the database to detect the active server. The polling interval is computed as follows:

$$\text{Polling interval} = \text{Database Update Frequency} \times \text{Database Polling Factor}$$

By default, the polling interval is 15 (update frequency) x 4 (polling factor) or 60 seconds. After 60 seconds, the monitoring server polls the database. If the monitoring server cannot detect the active server, failover occurs. You can configure the database update frequency and polling factor settings using server shared parameters in the Topology in Admin perspective.



## Failover

If both stage 1 and stage 2 monitoring fail, failover is automatic and it occurs as follows:

- The monitoring server performs a warm start.
- The monitoring server becomes the active server and takes over the workload.
- The new active server contacts agents to inform them to communicate with it.
- The new active server sends an SNMP message and an email to notify you about the failover and that the new active server is now running workload.

**Note:** Depending on the workload, the failover of workload can take 5-20 minutes. It may take longer even if the total number of active application generations is above 400 to 500. You must also plan for additional time for the scheduler to update all the agents.

## Failback

After a failover occurs, processing can resume on the failed server after it restarts (failback). You can configure failback to be automatic or manual. With automatic failback, the Primary is the preferred server. Workload processing is returned to the preferred server as follows:

- On starting, the preferred server sends a changerole request to the current active server.
- The currently active server quiesces (pauses) the server components and returns to monitoring mode.
- The preferred server becomes active and takes control of the workload; it also updates the agents

With manual failback, workload processing is not automatically returned to the failed server. When the failed server restarts, it runs in monitoring mode. To switch workload processing to the failed server, you must issue a `CHANGEROLE` command in the CLI on either server. After the `CHANGEROLE` command is issued, the current active server changes to monitoring mode and the current monitoring server becomes active.

# Chapter 6: Scheduling

To run workload, you need four main artifacts: an Application, Event, Jobs, and an Agent. The Application is a logical container of jobs that are executed by the agents. The jobs in the application may have dependencies to run in a specific order or concurrently.

## Applications

---

An Application consists of one or more jobs and other workload objects. Applications let you group related jobs and define their dependencies. For example, you may want to group all your sales department jobs in one Application. Jobs in an Application may run on different platforms.

## Jobs

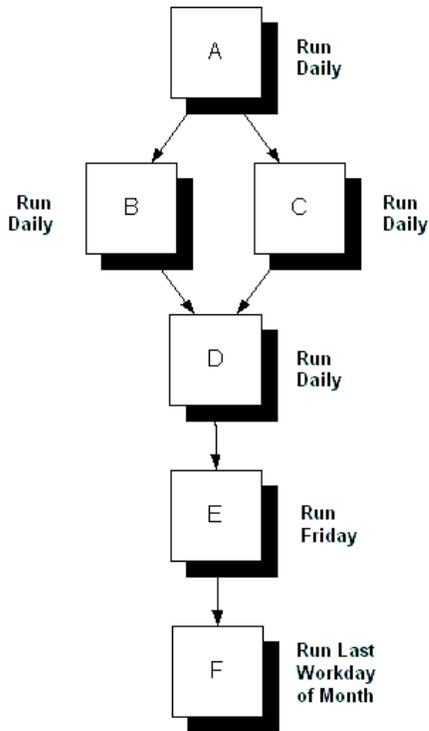
---

Jobs let you run programs, commands, or scripts on your operating system or on a specific application. Jobs are defined inside an application; different types of jobs pointing to different resources may be part of an application.

By default, the server submits a job after all of the job's predecessors complete successfully. Predecessors are workload objects that must complete before another workload object can run. Successors are workload objects that must wait for the completion of predecessors before they can run.

### Example: Predecessors and Successors in an Application

A sample application has six jobs (A, B, C, D, E, and F): The run frequencies of each job are as follows: A, B, C, and D run every day, E runs on Friday, and F runs on the last workday of the month. The following diagram illustrates the days on which the various jobs run:



The following table lists the predecessors and successors of each job:

Job	Predecessors	Successors
A	None	B, C
B	A	D
C	A	D
D	B, C	E
E	D	F
F	E	None

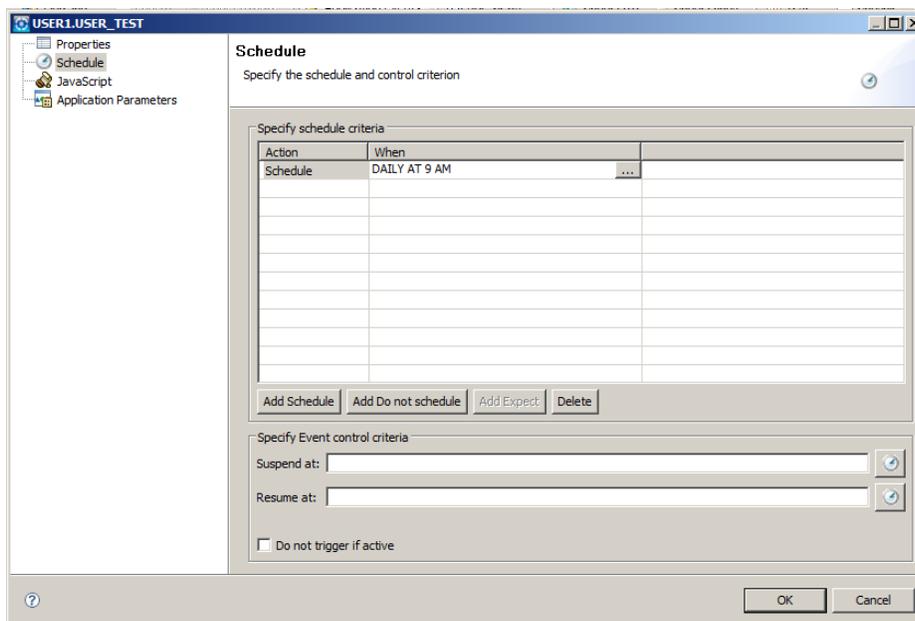
## Events

An event can be defined as "a change in state". For example, when a consumer purchases a television (TV), the TV's state changes from "for sale" to "sold". The TV shop's system considers this state change as an event whose occurrence can be made known to other applications.

You can use Events to run the workload defined in your Applications. When an Event is triggered, the Application runs. An application must have an event; a Date-Time event with no schedule is created by default.

Events can be scheduled, triggered manually, or triggered based on a particular condition.

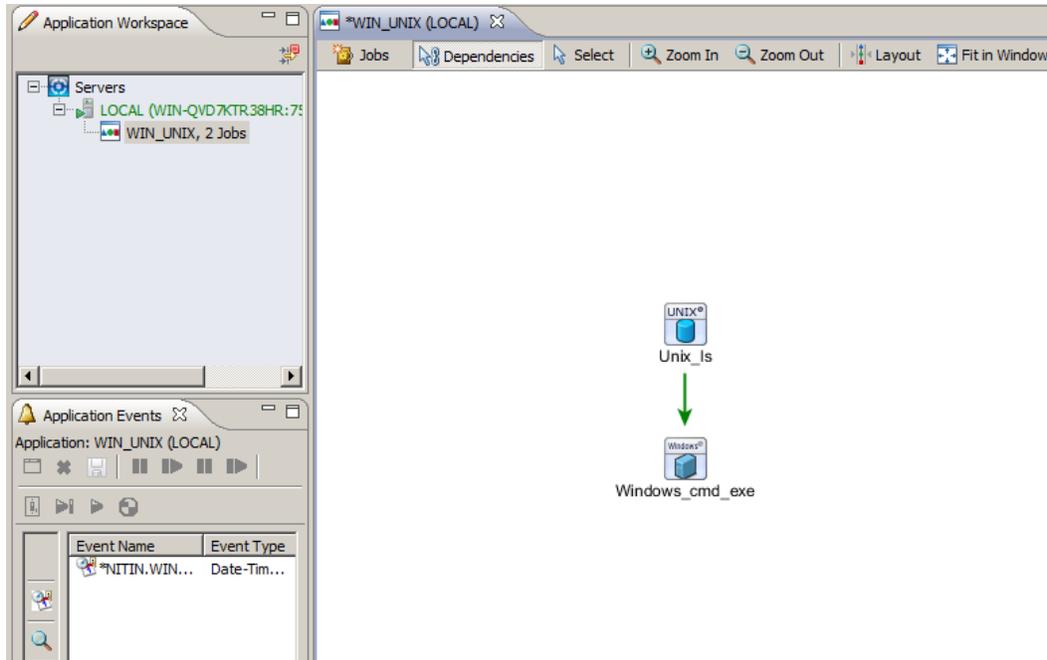
For example, you can schedule an Event to run an Application daily at 9 a.m. The following screenshot shows an example of an event with a schedule. The event will get triggered DAILY at 9 AM.



You can also trigger the Event manually if you want to add a new scheduled Event execution to the schedule or replace the Event's next execution. Once the scheduler triggers the event, the application is run and all the jobs within the application are sent to the Agent for execution.

This modular approach allows users to define complex workload easily and quickly. The users can easily add or edit the Jobs in Applications, change Events, modify Agents at the application or job level.

When an application is created, a default agent is also added in the definition. The jobs also require an agent, and by default they inherit the agent defined at the application level. However, the agent can be changed based on the requirement. For example, if you have an application with a UNIX job and a Windows job. The same agent is not applicable for both jobs. Each job can have its own agent.



## JavaScript

---

JavaScript is a scripting language commonly used to create dynamic HTML pages that process user input. With CA Workload Automation DE, JavaScript supports many functions that are unavailable from the CA WA Desktop Client interface. The name of each script must be unique for every instance of the server.

By using JavaScript scripts within your Applications, you can take advantage of the server's advanced scheduling features.

You can use JavaScript scripts to do the following:

- Create and manipulate symbolic variables
- Use server built-in functions
- Perform comparison, arithmetic, and logical operations
- Prepare program input and parameters
- Build decisions into schedules

**Note:** The server uses JavaScript Release 1.5 as its internal scripting language and conforms to Edition 3 of the ECMA-262 Standard for scripting languages.

You can also use a script to perform an operation within the Alert definition. You can define the script in the Alert definition or store it in the JavaScript repository.

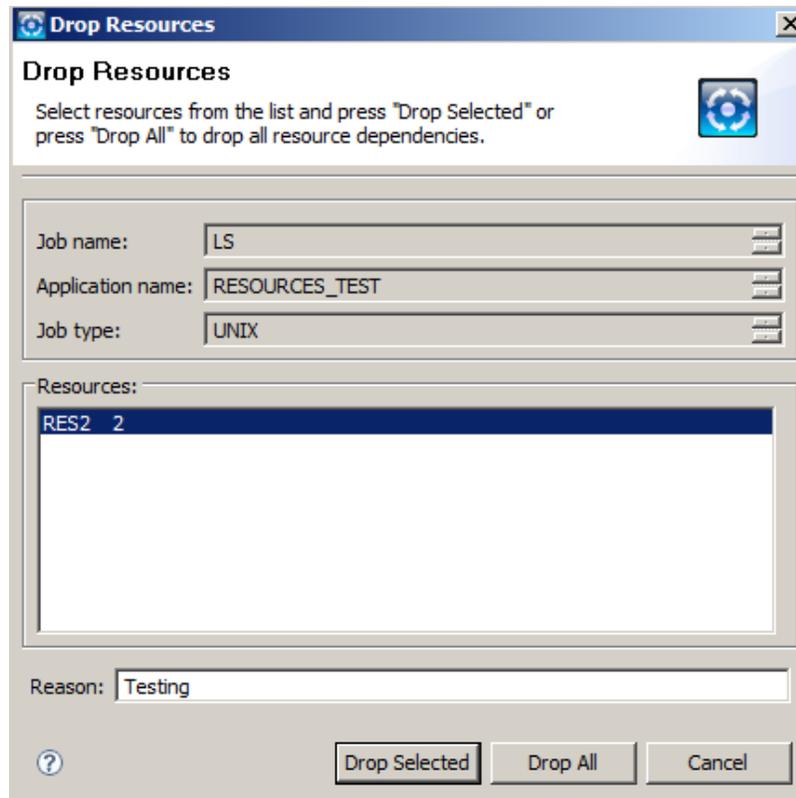
## Resources

A resource is a job dependency that can be quantified by specifying its availability count. Resource dependencies affect a job's ability to run successfully. The server submits only jobs that have met all their resource requirements. You can specify one or more resources to jobs. When a job is waiting for resources, it goes into a RESWAIT state.

For example, if a job needs two units of a resource and only one unit is available, the job cannot run (RESWAIT) until all units of the resource are available.



You can also release the resources dependencies if you no longer require the job to wait. To release the resource dependency, navigate to Monitor perspective and select the job in RESWAIT state. Right-click the job and select Drop Resources to drop the selected resources or all resources.

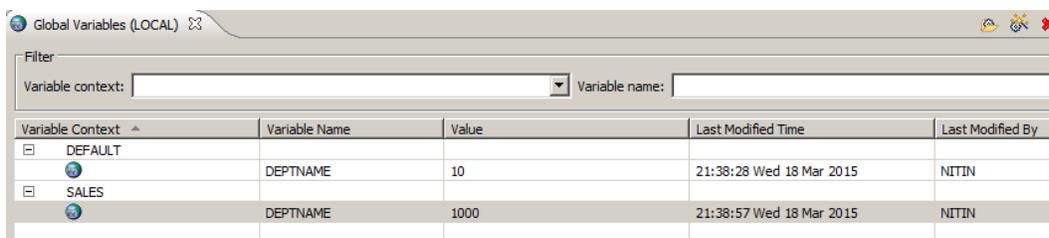


**Note:** Resources let users easily control the workload. As your workload increases, the usage of number of resources may also increase. We recommend that you look into other CA Workload Automation solution when the number of defined resources is increased to 2000 or more. Global Variables are another alternative to resources and is also efficient.

## Global Variables

Global variables let you store information that you can reuse across Applications. Global variables save time; you do not have to enter specific information, such as job names or argument values, each time you want to perform the same type of processing. When you use global variables, you also reduce the possibility of coding errors. These variables can be created, modified, and deleted from the Services perspective in CA WA Desktop Client, the CLI (Command-Line Console), or using JavaScript functions.

Each global variable belongs to a context, which is a group of related variables. Contexts help you avoid naming conflicts. For example, you can create two variables named DEPTNAME, each in a different context. By default, all global variables are defined in the DEFAULT context.



Variable Context	Variable Name	Value	Last Modified Time	Last Modified By
DEFAULT	DEPTNAME	10	21:38:28 Wed 18 Mar 2015	NITIN
SALES	DEPTNAME	1000	21:38:57 Wed 18 Mar 2015	NITIN

You can use global variables when you define jobs. The %VAR statement lets you specify a global variable name in supported job definition fields. When an Event is triggered, the server substitutes the current value of that global variable.

Syntax:

```
%VAR('name' [, 'context'])
```

### Example:

To specify the global variable in database jobs, enter the following %VAR statement in the DB URL field in the Basic page of the job definition dialog:

```
%VAR('dburl', 'dbapps')
```

## Calendars

CA Workload Automation DE has many integrated scheduling terms, such as dates and times. You can set up additional scheduling terms, such as holidays, Thanksgiving, special days, and special processing periods, which are unique to your environment. You can define these terms in CA Workload Automation DE using a calendar.

You can define calendars to meet your installation requirements. The administrator controls access to these calendars. A calendar contains the following:

- Holidays (holidays can span one day or several days)
- Special days such as inventory day, company meeting, or pay day
- Special processing periods such as fiscal months, fiscal years, and 4-5-4 periods
- Days of the week considered to be workdays, for example, Mon-Sat or Mon-Fri

## Application LifeCycle

---

CA Workload Automation DE is multithreaded. The server has several components each running its own thread. The Scheduler thread is for initiating the Application lifecycle. When an event is triggered based on its schedule, the Scheduler thread loads the Time Driven Request (TDR). The TDR is an XML representation of the Event.

A sample server tracelog is as follows:

```
20150318 00:58:22.682 [essential] [INFO] SS: Processing TDRs having execution
time = 1426654702597; Wed Mar 18 00:58:22 EDT 2015
```

```
20150318 00:58:22.920 [essential] [INFO] SS.tdr-executer: Executing event
USERA.ECHO_JOB_2015-03-18_00:58:22.597_F2CBC3F9-EEE9-4EF6-96F2-8CACA7987
```

The scheduler processes JavaScripts (if defined) and then sends the application definition to the scheduling manager. The scheduling manager then builds the new TDR for the next schedule and updates the Event. For example, you defined a schedule “DAILY AT 9 AM”. At 9 AM, the server executes the current TDR and builds the new one for the next day. You can see the date and time for the next event trigger in the Events under the Services perspective. The ‘Next Scheduled Time’ column shows the date and time when the event will trigger. This column is updated each time the event is triggered.

You can specify JavaScript to be executed at Event trigger time in the Application. You can only run scripts that are defined in the JavaScript repository. Select the script from the drop-down list as shown below:



Once the scheduling manager or Distributed Manager (DM) thread receives the Application definition, it builds the jobs definitions into the memory. It will also execute any JavaScripts that may have been defined. The scheduling manager then sends the jobs to the agents.

A sample Application that is being built and started is as follows:

```
20150318 01:17:35.736 [trace:afm] [INFO] DM.Appl.ECHO_JOB.4: <-- 20150318
01173563+0400 DE_7500 SCHEDULER /MAIN/ECHO_JOB.4/# Appldef End
Userid(SCHEDULER) 20150318 01:17:35.736 [essential] [INFO] DM.Appl.ECHO_JOB.4:
nonApplMgr.getDistantWobs(ECHO_JOB,ECHO) returns{}

20150318 01:17:35.754 [essential] [INFO] DM.Appl.ECHO_JOB.4: Application
ECHO_JOB.4.ECHO_JOB~~ has started.
...
```

The scheduling manager then waits for the agent to send updates on the job. It processes the information and store in the database. If the Application completes, it terminates the thread and updates the database. The data is also stored in the history tables for any reporting or auditing purposes.

A sample entry of the ECHO\_JOB from ESP\_APPLICATION table is as follows:

JOB_NAME	APPL_NAME	START_DATE	END_DATE	SCHEDULED_DATE	TIME	STATE	EVENT_NAME
ECHO_JOB~~	ECHO_JOB	18-MAR-15	18-MAR-15	18-MAR-15	COMPLETE	USERA.ECHO_JOB	

More job-specific details are stored in the ESP\_GENERIC\_JOB table.

## Job LifeCycle

---

When an application is created, as described in the Application LifeCycle section, the jobs come in to existence. A job may have several requirements, such as resources, time constraints, and predecessor jobs. The server has to wait for all the job requirements to complete. When all the conditions are met, the job is considered runnable. The server then sends the Automated Framework Message (AFM) to the agent.

A sample AFM that is sent to the Agent (AGENT007) is as follows:

```
20150318 01:17:36.159 [trace:afm] [INFO] DM.Appl.ECHO_JOB.4: --> 20150318
01173605+0400 AGENT007 DE_7500 /MAIN/ECHO_JOB.4/ECHO RUN . Data(Args="/c
C:\\TEMP\\echo.bat hello",Command=cmd) User(administrator)
Password(XXXXXXXXXXXXXXXXXXXXXXXXXXXX) TargetSubsystem(WIN) MFUser(USERA)
```

The agent accepts and acknowledges the AFM. The agent then sends update on various job states to the scheduling manager, for example, EXEC, COMPLETE or FAILED.

A sample Agent EXEC update on the job (from server tracelog) is as follows:

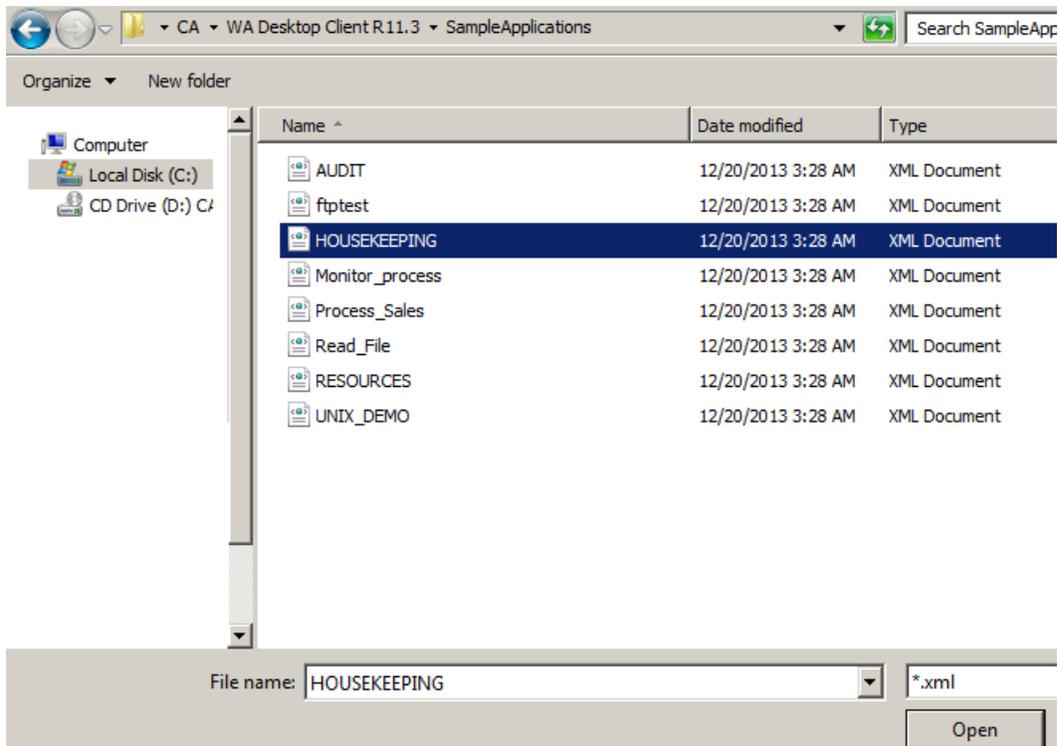
```
20150318 01:17:36.710 [agent:afm] [INFO] DM.InputProcessor_12: <-- 000000001653
000000001654 == true 20150318 01173663+0400 DE_7500 AGENT007
ECHO/ECHO_JOB.4/MAIN State EXEC SetStart Status(Executing at AGENT007)
Jobno(4128) JobLogId(5D5D3153E5689C686592488845CA9173190E68A4)
User(AGENT007) Host(WIN)
```

# Chapter 7: Administration

The CA Workload Automation DE contains several built-in utilities for day-to-day maintenance. You must schedule these utilities as part of the workload and run them daily.

## Housekeeping Application

CA WA Desktop Client is packaged with a sample Application named HOUSEKEEPING. This application is located in the <drive:>\<WA Desktop Client install>\SampleApplications directory. You can use this application to perform various housekeeping operations. It must be scheduled to run daily during non-peak hours.



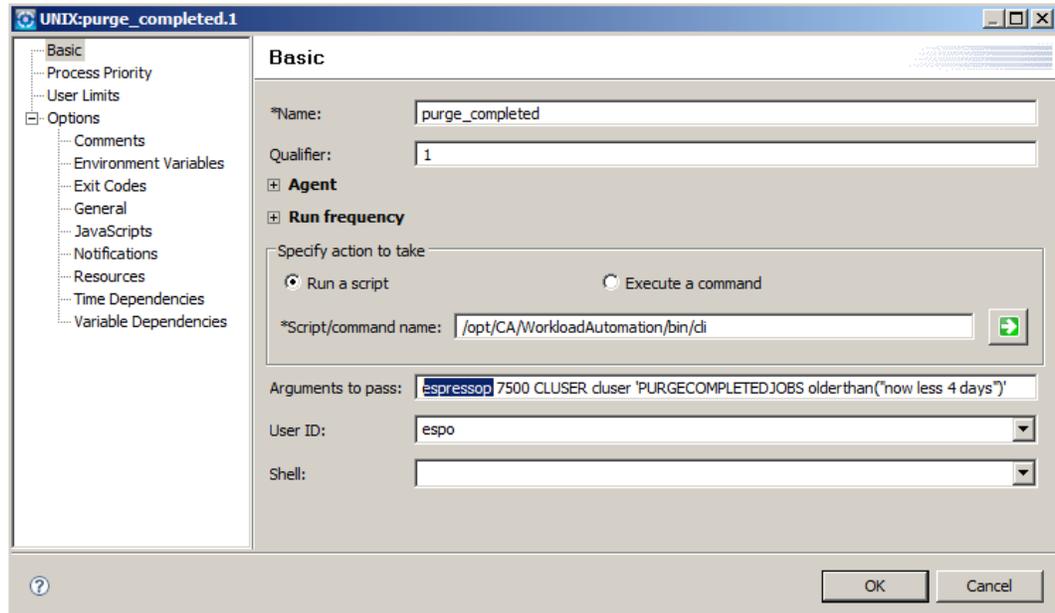
The housekeeping Application contains the following jobs:

### cleanup

Removes archived log files older than a specified number of days.

### purge\_completed.1

Clears all completed jobs that are older than a specified number of days from the server completed jobs repository.



Ensure that the 'Script/command name:' matches the server installation directory and 'Arguments to pass:' matches the hostname, port, user, and password.

### purge\_completed.2

Clears all completed jobs in a specified Application generation from the server completed jobs repository. This job runs on request.

If you are working on a UNIX environment, you can modify and schedule this Application to perform routine maintenance of your server.

**Note:** On Windows environment, use the HOUSEKEEPING Application as a template and change the job types to Windows.

## PURGELOG

You can clear server log files by issuing the PURGELOG command in the CLI. The syntax of this command is as follows:

```
PURGELOG [AGE (age) ]
```

### Examples:

```
purge log age(3)
purge log
```

Alternatively, you can clear server log files using the cleanup.sh or cleanup.bat script that is packaged with the server. It can be added as UNIX or Windows job in the HOUSEKEEPING application. The syntax of the script is:

## Workload Maintenance

### Windows:

```
cleanup.bat server_host server_port user password days
```

### UNIX:

```
cleanup.sh server_host server_port user password days
```

### Examples:

#### Windows:

```
C:\Program Files\CA\Workload_Automation_DE_11_3\bin>cleanup.bat localhost  
7500 admin admin 5
```

#### UNIX:

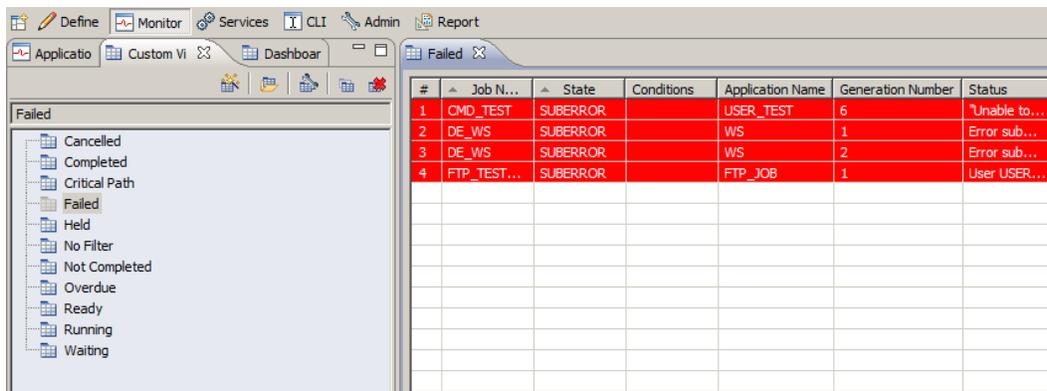
```
./cleanup.sh localhost 7500 admin admin 5
```

## Workload Maintenance

---

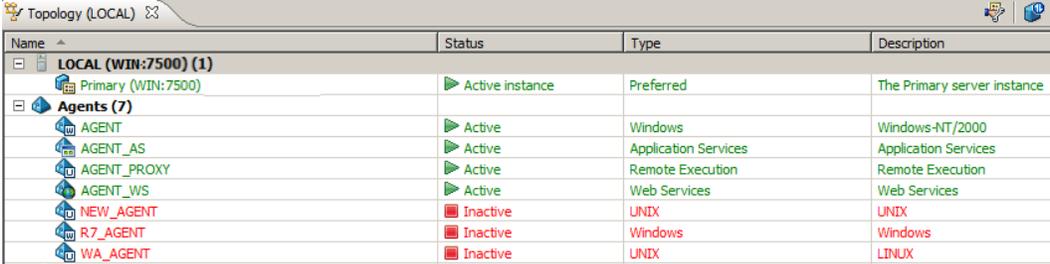
In a busy environment, the jobs may fail from time to time. The server has to spend more resources to track the FAILED jobs. This can be significant if there are hundreds of FAILED jobs. Anytime, an application that is not COMPLETE is considered ACTIVE. An Application with FAILED job is considered active. By default, the server is configured for 600 active generations. If it reaches above 600, the server will shut down.

**Important!** Clear FAILED jobs and applications by FORCE completing them or by fixing the issues that are associated with the jobs. Use the 'Failed' custom view in the Monitor perspective for listing failed jobs.



#	Job N...	State	Conditions	Application Name	Generation Number	Status
1	CMD_TEST	SUBERROR		USER_TEST	6	Unable to...
2	DE_WS	SUBERROR		WS	1	Error sub...
3	DE_WS	SUBERROR		WS	2	Error sub...
4	FTP_TEST...	SUBERROR		FTP_JOB	1	User USER...

One of the most common reasons for a job failure is INACTIVE agents. If the agents remain down or inactive for a prolonged period, the jobs associated with them stacks up. The Admin perspective shows all the agent states under Topology as shown below:



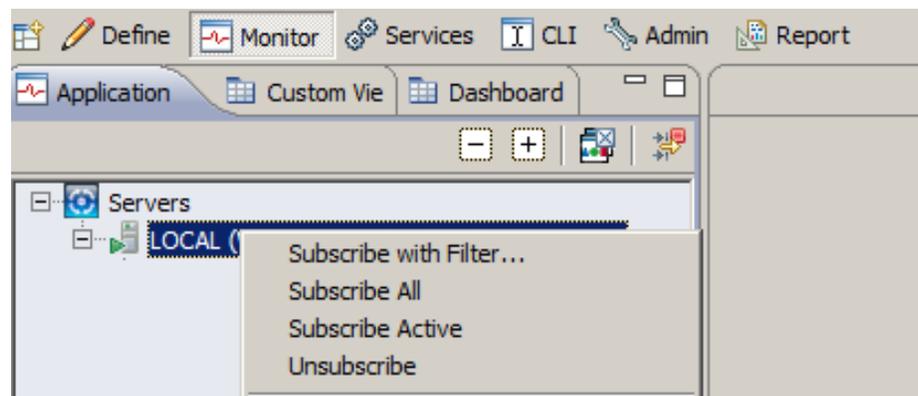
Name	Status	Type	Description
LOCAL (WIN:7500) (1)			
Primary (WIN:7500)	▶ Active instance	Preferred	The Primary server instance
Agents (7)			
AGENT	▶ Active	Windows	Windows-NT/2000
AGENT_AS	▶ Active	Application Services	Application Services
AGENT_PROXY	▶ Active	Remote Execution	Remote Execution
AGENT_WS	▶ Active	Web Services	Web Services
NEW_AGENT	■ Inactive	UNIX	UNIX
R7_AGENT	■ Inactive	Windows	Windows
WA_AGENT	■ Inactive	UNIX	LINUX

You must either delete the inactive agents if they are no longer needed or fix the issues related to the agent down or inactive state. The most common reasons for an agent to appear inactive are as follows:

- Network connectivity; the port might have been blocked
- Agent has been shut down and needs an administrative action
- Agent had been misconfigured, for example, incorrect IP and port might have been specified in the agent definition

## Monitoring Considerations

When large numbers of clients are monitoring workload, you can use the 'Subscribe Active' subscription option to subscribe active applications and jobs. This option is ideal for daily and constant monitoring. This option is more efficient and faster than other options because it reads the active generation list from the application memory.



The 'Subscribe all' or 'Subscribe with filter' subscription options may only be used when extensive or information about certain application is needed. The 'Subscribe all' or 'Subscribe with filter' are resource-intensive. They require database scanning to gather all the applications. When there are more than 100 concurrent Client connections, the 'Subscribe Active' should be used more often.

## Minimum Free Memory

The server has several settings to alarm users when it runs low on heap or memory. The default memory for an alarm to execute is 200 KB (204800 Bytes). The server executes a 911 alarm if the memory is 200 KB or below. The server starts shutting down if the total available memory is 100 KB (102400 Bytes) or less.

Increase these default values in the Production setup by changing the value of the following server shared parameters:

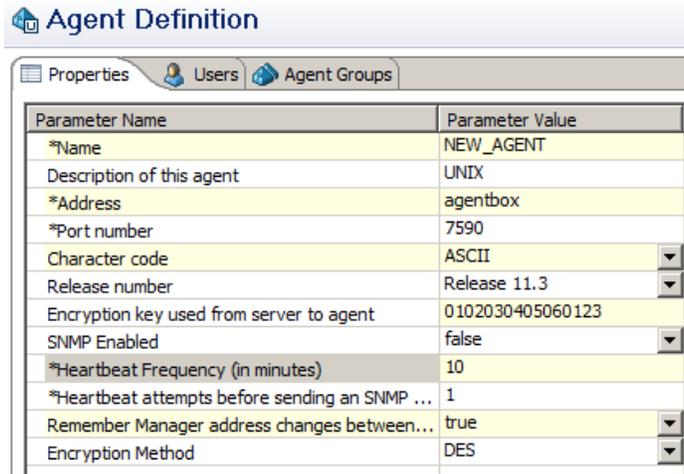
\*Minimum free memory before 911 executes an alarm action (bytes)=&\$- +%) &\$S` fI&\$S` A6L

\*Minimum free memory before 911 executes a critical action (bytes)=%&,(, ) +\* \$S` fI%\$S` A6L

Parameter Name	Parameter Value
*Delay spool file retrieval for email attachment (seconds)	30
*Maximum length of JavaScript	65536
*Manager thread wait time for input queue (milliseconds)	300000
*How often to check for orphan report documents (milliseconds)	60000
*How often a client reads from a remote stream before is stuck (millise...	120000
*How many scheduled reports can be run at the same time	4
*How many users can run reports at the same time	4
*How long to wait for a report to finish (milliseconds)	900000
*Scheduler thread wait time for input queue (milliseconds)	300000
*Variable manager thread wait time for input queue (milliseconds)	300000
*Tickle message priority	0
*APPLUMP message priority	0
*Minimum number of threads per pool	1
*Maximum number of threads for all thread pools	500
*Heartbeat interval for the timerpop (milliseconds)	3600
*Interval for inspecting spool file requests (milliseconds)	30000
*Global agent heartbeat interval in minutes (0 to disable)	0
*Amount of time to delay an agent heartbeat (milliseconds)	250
*Maximum agent heartbeat ping wait time (milliseconds)	1000
*Maximum agent response read wait time (milliseconds)	60000
Propagate APPLHOLD and APPLQUIESCE status to jobs	true
Disable the 911 service	false
*Maximum threads before 911 executes an alarm action	2500
*Maximum threads before 911 executes a critical action	3000
*Minimum free memory before 911 executes an alarm action (bytes)	204800
*Minimum free memory before 911 executes a critical action (bytes)	102400
*Acceptable error frequency before 911 goes from alarm to critical state	0.1

## Agent Heartbeat

The CA Workload DE server uses heartbeat or special ping to check on all the agents. If the agent does not respond back, it is marked as INACTIVE in the Topology. The default heartbeat ping is 5 minutes. High Heartbeat frequency may sometimes cause additional congestion in your network. In a Production environment, this can be increased to 10-15 minutes for each agent.



Parameter Name	Parameter Value
*Name	NEW_AGENT
Description of this agent	UNIX
*Address	agentbox
*Port number	7590
Character code	ASCII
Release number	Release 11.3
Encryption key used from server to agent	0102030405060123
SNMP Enabled	false
*Heartbeat Frequency (in minutes)	10
*Heartbeat attempts before sending an SNMP ...	1
Remember Manager address changes between...	true
Encryption Method	DES

However, if there is an issue with an agent, you can decrease the Heartbeat Frequency to 5 minutes or below. You cannot set the frequency below 1 minute.



# Chapter 8: Database Requirements

CA Workload Automation DE requires access to an RDBMS system. The database serves as a repository of server configuration files, resource definition files, and historical reporting data. It requires its own schema or user with full access to a dedicated tablespace (or Database in MS-SQL).

The scheduler uses Java Database Connectivity (JDBC) to connect to the database. JDBC is an object-oriented database call-level interface for Java specified by Sun Microsystems. Applications written in Java can access databases using the JDBC API. A native JDBC driver is delivered with the product.

The CA Workload Automation DE r11.3 supports the following databases and versions:

Database	Version
Oracle	10gR1 and R2, 11gR2 and RAC, and 12c
Microsoft SQL Server	2005, 2008, and 2012
IBM DB2	9.1, 9.5, 9.7, 10.1 <b>Note:</b> DB2 option is only available when CA Workload Automation DE is installed on AIX, or z/LINUX. However, DB2 database may be installed on any vendor-supported platform such as AIX, Windows, or z/OS.

## Database Setup

---

For Oracle database, set up a schema with a default tablespace. This tablespace must not be shared with other external applications. A sample schema and tablespace setup SQL script is as follows:

```
CREATE TABLESPACE de_tablespace
DATAFILE '/opt/app/data/DE/DE_TABLESPACE_01.dbf'
SIZE 2000M
AUTOEXTEND ON
NEXT 200M
MAXSIZE 20000M;
---Optional MAXSIZE, you can set to unlimited
CREATE USER deuser IDENTIFIED BY some_passsword
DEFAULT TABLESPACE de_tablespace
TEMPORARY TABLESPACE temp
QUOTA UNLIMITED ON de_tablespace;

GRANT CONNECT TO deuser;
GRANT CREATE SESSION, SELECT ANY TABLE, UPDATE ANY TABLE, INSERT ANY TABLE TO deuser;
GRANT RESOURCE TO deuser;
----OPTIONAL GRANT----
GRANT select_catalog_role to deuser;
```

For Microsoft SQL database, a dedicated user and a mapped database must be used. The mapped user must have db\_owner or similar privileges to the database. Any other application or user must not share the database.

## Database Maintenance

---

As the workload increases, the tables may require additional maintenance to prevent any performance degradation. Since the CA Workload Automation DE server stores all the artifacts, definitions, and other historical information, we recommend that you run periodic maintenance on the database.

CA Workload Automation DE r11.3 is packaged with built-in utilities for table cleanup. The utilities can be run manually from CLI or can be executed as a part of workload without the need for a scheduler shutdown.

### PURGECOMPLETEDJOBS

When the server runs, it collects and stores information about active and completed jobs in its relational database. The server completed jobs repository in the database stores information about completed jobs. To maintain the database performance, you must periodically clear the server completed jobs repository by issuing the PURGECOMPLETEDJOBS command. The command can be defined as a CLI job and ran daily. We recommend that you keep the history data of completed jobs at 14 days or less, or below 100 generations for cyclic jobs.

**Examples:**

```
PURGECOMPLETEDJOBS OLDERTHAN ("NOW LESS 14 DAYS")
PURGECOMPLETEDJOBS APPLICATION("MY_APPLICATION")
```

The olderthan <string> purges jobs older than the specified scheduled time. The application <string> purges the specified application generation and its predecessors. You can also define regular expression to match certain application names, for example, "all|[a-zA-Z0-9@\$\_\*?]+(\.\d+)\*".

```
PURGECOMPLETEDJOBS application("A*")
```

**MOVEHISTORYDATA**

The MOVEHISTORYDATA command moves historical data in the database tables (ESP\_APPLICATION and ESP\_GENERIC\_JOB) to the corresponding H or staging tables within the same tablespace. These tables store history data of all the jobs that have ever executed. When you run an Application, the ESP\_APPLICATION table stores the information about the application automatically. The ESP\_GENERIC\_JOB stores the information about each job.

Over time, the table records can increase to millions. These tables should be kept at or below one to two million records. Depending on the workload, you may need to move data older than 30-90 days. This command can be executed daily and can be a part of workload.

**Example:**

```
MOVEHISTORYDATA OLDERTHAN ("TODAY LESS 2 MONTHS")
```

A sample script that can be executed as a command job is as follows:

UNIX:

```
./cli de.example.com 7500 user MY_PASSWORD MOVEHISTORYDATA OLDERTHAN("\TODAY LESS 2 MONTHS\")
```

Windows:

```
cli.bat LOCALHOST 7500 user MY_PASSWORD MOVEHISTORYDATA OLDERTHAN("\TODAY LESS 2 MONTHS")
```

The command may require escaping of special characters by “\” as shown in the above examples.

**DELETESTATUSMESSAGES**

CA Workload Automation DE r11.3 has a feature that lets users create new status views in the Monitor perspective. You can create a new status message view to select the message types that you want to display in the new view. You can also specify the message sorting order of the new view. The server stores this data in several tables, and these tables may require a routine cleaning.

The DELETESTATUSMESSAGES CLI command deletes historical status messages that are created before a specified date. The following message types are considered historical: SNMP, SNMP application, login failure, event authorization, and log message.

**Example:**

```
DELETESTATUSMESSAGES
DELETESTATUSMESSAGES THRESHOLD("2010-03-17 10:00:00")
```

The command can be issued with no switches or with a date threshold. If no argument is provided, the command will delete all the status messages in the status message tables.

When you run the command for the first time, we recommend that you run the below query and determine the total number of records to delete from the ESP\_STATUS\_MESSAGE table:

```
SELECT COUNT(*) FROM ESP_STATUS_MESSAGE where CREATED_DATE <
TO_DATE('2014-06-29', 'YYYY-MM-DD');
```

If the number of records is greater than 2 million, modify the THRESHOLD date accordingly. Specify the oldest date and increment the threshold date one day at a time. If the database keeps the timestamps to a precision of up to 1/100 of a second, it can be useful when deleting large of number of rows in smaller chunks.

```
DELETESTATUSMESSAGES THRESHOLD("2014-03-01 10:00:90")
```

**Increase Database Connections**

When the number of jobs and applications increases, the server requires more database connections for timely updates. The server has a setting that determines how many active connections it can open with the database. The default number of active database connections is 50.

**Important!** In a busy server environment, we recommend that you increase the simultaneously open database connections to 400 or more. After the number of connections is set on the server side, the database change is also required. The database must also allow the number of connections set on the server side.

The setdbparm command in <CAWA install directory>/bin directory is used for changing all database-related settings. The command changes the values in the db.properties file, which is located in <CAWA install directory>/conf directory.

**Example: Change the Maximum Database Connections Settings**

```
setdbparm.bat database.maxconnections.in.pool
Changing max connections in pool (database.maxconnections.in.pool)
Enter new max connections in pool: 100
Verify new max connections in pool: 100
Changed. The Server must be restarted for the change to take effect.
```

## Idle Connections

Databases, such as Oracle, may mark a session as INACTIVE when it is idle or has completed all the transactions. However, these sessions will remain connected to the database. In Oracle, the status can be checked in `v$session`. In a busy environment, the idle connections may sometimes need to flush more rapidly to allow new connections. The default release time for an idle connection is 1800 seconds. To release the unused database connections, reduce the idle connection time to 1200 or 600 seconds by using the `setdbparm` utility:

```
setdbparm database.connection.idle.flush.threshold
Changing idle flush threshold (database.connection.idle.flush.threshold)
Enter new idle flush threshold: 1200
Verify new idle flush threshold: 1200
Changed. The Server must be restarted for the change to take effect.
```

## Database Tuning

---

Apart from regular database maintenance, additional maintenance may also be helpful. Some common Oracle and MS-SQL recommendations.

### Oracle Database Considerations

The Oracle database has several built-in mechanisms that improve the overall database performance:

- Contact your database administrator to partition large tables and improve the database performance.
- For `ESP_GENERIC_JOB` & `ESP_APPLICATION` tables, change the value of `INITTRANS` to 20.
- Increase the CLOB chunk size to 16 KB for the all table that use of CLOB or BLOB data type:
  - `ESP_AMS_DF`
  - `ESP_AMS_VR`
  - `ESP_APPLICATION`
  - `ESP_EVENT_DF`
  - `ESP_EVENT_RP`
  - `ESP_JOB_AETDATA`
  - `ESP_L2O_MAP (BLOB)`
  - `ESP_MANAGER_INQ`
  - `ESP_MANAGER_OUTQ`
  - `ESP_RM_INQ`
  - `ESP_RT_WOB (BLOB)`
  - `ESP_S2O_MAP (BLOB)`

- ESP\_SS\_ACTION\_INQ
- ESP\_SS\_INQ
- ESP\_SS\_REPORT\_RESULT (BLOB)
- ESP\_SUBSCRIPTION\_CMD (BLOB)
- ESP\_TDR\_DATA
- ESP\_VAR\_INQ
- ESP\_WSS\_JOB
- H\_APPLICATION

### SQL Server Considerations

By default, SQL Server uses optimal configuration settings when you install it. If you change the configuration settings, ensure that you carefully test the configuration changes before applying them in the production environment. Some important recommendations for SQL server are as follows:

- Relocate the logs and the tempdb database on separate devices from the data
- Provide separate devices for heavily accessed tables and indexes, such as ESP\_GENERIC\_JOB and ESP\_RTWOB
- Use the appropriate RAID configuration
- Pre-grow the databases and logs to avoid automatic growth and fragmentation performance impact
- Maximize the available memory
- Manage the index fragmentation
- Perform regular database administrative tasks

# Chapter 9: LDAP

CA Workload Automation DE supports Lightweight Directory Access Protocol (LDAP) authentication. LDAP is a powerful protocol for accessing directory services. It lets users easily search, retrieve, and manipulate directory content.

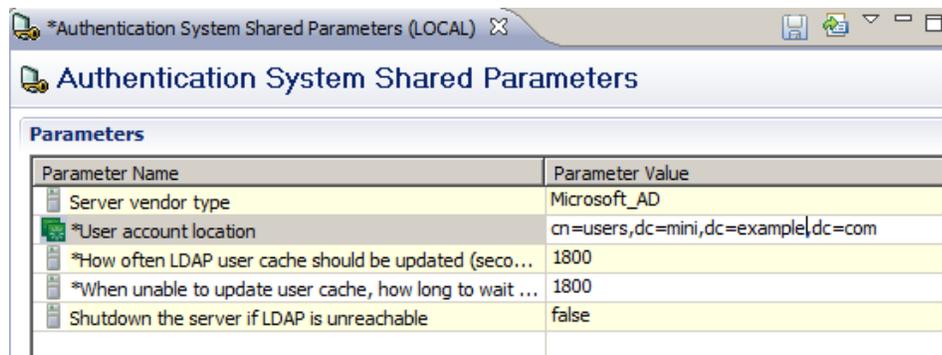
The CA Workload Automation DE server acts as an LDAP client. It authenticates (binds) and retrieves the users.

The following LDAP vendors are supported:

- Microsoft Active Directory
- Novell Directory Services
- Sun Directory Services

The LDAP settings can be configured during the installation. It can also be added or edited later after the installation has been complete. When setting up an authentication system in CA Workload Automation in DE, commonly used set of shared parameters have to be added.

The value of the User Account Location parameter must be the fully qualified path where the Users are located. This value is used as a 'Base DN' to retrieve users either from one-level or from sub-tree. CA Workload Automation DE supports specifying only an Organization-Unit as 'User account location' from where the users are retrieved.



Parameter Name	Parameter Value
Server vendor type	Microsoft_AD
*User account location	cn=users,dc=mini,dc=example,dc=com
*How often LDAP user cache should be updated (seconds)	1800
*When unable to update user cache, how long to wait ...	1800
Shutdown the server if LDAP is unreachable	false

**Important!** By default, LDAP user names are retrieved and cached every 30 minutes by the CA Workload Automation DE Server. We recommend that you increase the value of "How often LDAP User cache should be updated (seconds)" to 12 hours (43200 seconds) to once a day (86400 seconds). Normally, in a production environment, the directory services may not see frequent daily updates.

**Note:** The CA Workload Automation DE Server can be configured to shut down if the LDAP server is not reachable. You can configure it by setting "Shutdown the server if LDAP is unreachable" to true. We recommend that you do not change the default setting (false) unless LDAP requirement is extremely critical for various other organizational requirements.

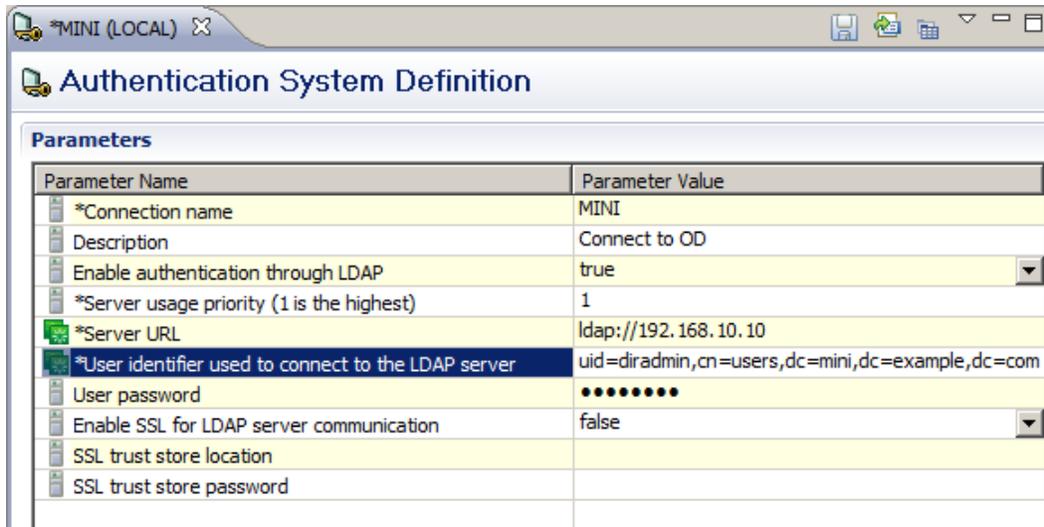
After the System Shared Parameters are defined, you can add an LDAP server. To add an LDAP server, the following information is required from the Domain or LDAP administrator:

- User with admin access/privileges that can bind the CA Workload Automation DE server to the directory server and also retrieve users from the 'User account location' specified in the Shared Configuration parameters settings. Define full path of the User.
- Exact server name or IP address of the directory server. For normal access, the protocol that is used is "ldap://" and for SSL access use "ldaps://"
- (Optional) Port, only if default is not used
- For SSL communication, the truststore file that contains the LDAP server certificate and the password that was specified while generating the truststore file.

**Note:** For more information about configuring CA Workload Automation DE server with the LDAP server, refer to this URL:

[https://support.ca.com/cadocs/7/CA%20Workload%20Automation%20DE%20r11%203%20SP3%20-%20Customer%20Access%20Only-ENU/Bookshelf\\_Files/HTML/Admin\\_Perspective\\_Help/index.htm](https://support.ca.com/cadocs/7/CA%20Workload%20Automation%20DE%20r11%203%20SP3%20-%20Customer%20Access%20Only-ENU/Bookshelf_Files/HTML/Admin_Perspective_Help/index.htm)

The following screenshot shows the correctly configured settings for adding an LDAP server to the topology. The LDAP user 'diradmin' has privileges to retrieve users from LDAP server. The password field has the password of 'diradmin'.



**Note:** Spaces and special characters are not allowed in the 'Connection name' field.

The “User identifier used to connect to LDAP server” field defines the exact user location. This location can be obtained by accessing the services using an LDAP browser or from your Directory administrator. Users can use any LDAP browser such as JXplorer, Thunderbird, and LDAPAdmin to retrieve location information. If you change the value of “User identifier used to connect to LDAP server”, you must also update the User account location in the “Authentication System Shared Parameters” settings.

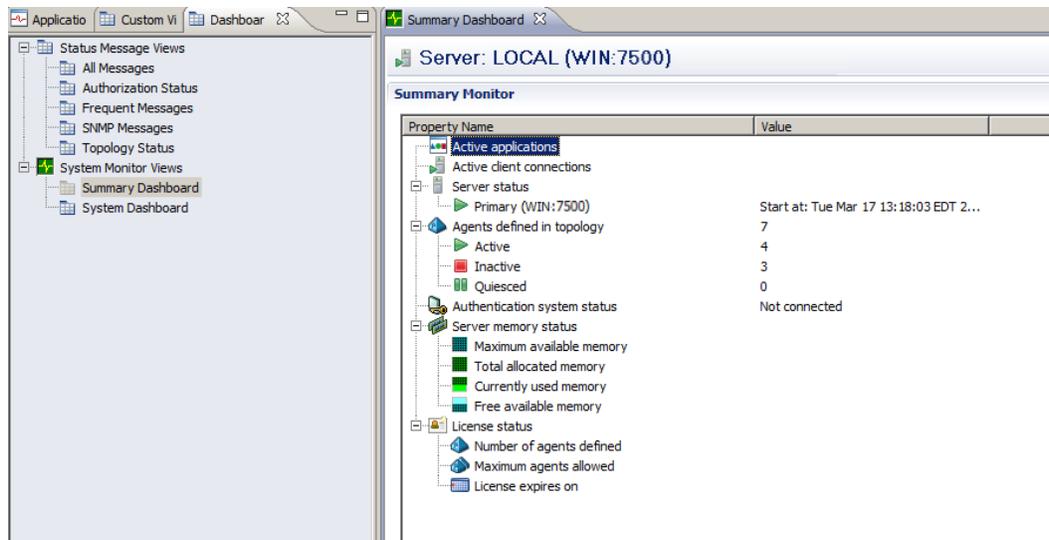
## Multiple Authentication Systems

---

The ‘Server usage priority’ is used when more than one LDAP Servers are configured with CA Workload Automation DE server. The CA Workload Automation DE server uses the LDAP Servers based on the priority that you specify. When LDAP server 1 is reachable, the CA Workload Automation DE server uses it to retrieve users. If the connection to LDAP server 1 fails, the CA Workload Automation DE server will try to connect to the next LDAP server, and so on, until a successful connection is made.

# Chapter 10: Additional Considerations

The Monitor perspective in CA WA Desktop client contains a Dashboard view that provides status and historical information about the server environment. The summary and system dashboards provide real-time information about the server environment setup.



The following recommendations are for the server:

- For better performance, the server must have at least 10-15 percentage of free memory available out of the total memory
- If the number of active applications is greater than 200 constantly, check for the failed jobs, long running applications, and the jobs in MANHOLD state and fix the issues associated with them.
- When the server authenticates users against LDAP, the 'Authenticated System status' field must be active.
- Number of inactive agents must be 0 or less than 5.
- The server has sufficient disk space. You can use local or default agent for 'Disk Monitoring' job.
- Set up SNMP traps and alerts for server disk space issues
- Ensure that the database has sufficient disk space for transaction logs and tablespace

## Agent Groups

---

In r11.3, you can group system agents (Windows or UNIX/Linux agents) of same operating system and load balance the jobs across the agents in the agent group. Each group can only have one type of agents; you cannot group UNIX agent and Windows agent in a group.

The server supports the following three types of 'Selection Criteria' for agent groups:

- CPU
- Round-Robin
- Random

The server uses CPU criteria to determine the CPU usage on each agent in the group. The CPU usage is determined by a selection monitoring done on the agent. The agent monitors the CPU load and sends update to the server every 5 minutes (default: 300 seconds). The server then collects the load information from all the agents and determines the best agent for running the next job.

**Important!** When defining agent groups, use agents that are installed on similar operating system, hardware, CPU and memory configuration. We recommend that you use the round-robin or random criteria instead of the CPU selection criteria.

If you use the CPU criteria, we recommend that you define the agent group with fewer than 10 agents or at most 15 agents. Use round-robin if you have more than 10 agents in a group.

## Implementation Checklist

---

The following checklist contains installation guidelines and recommended best practices:

Type	Recommended Settings	Enter Your Settings
<b>Installation</b>		
OS Bit Type	64	
CA Workload Automation DE Installation Bit Type	64	
Heap (Max 2 GB for 32 bit) (GB)	4	
Disk space (free in GB)	40	
Local Disk space monitoring (CA WA agent or other SNMP service)	Yes	
<b>Performance</b>		
PURGELOG	DAILY command job	
Increase free memory before 911 executes an alarm:		

Type	Recommended Settings	Enter Your Settings
Minimum free memory before 911 executes an alarm action (bytes)	26214400	
Minimum free memory before 911 executes a critical action (bytes)	13107200	
Agent Heartbeat (minutes)	10	
Logging Profile	ESSENTIAL	
No. of Inactive agents	0	
Latest Patch and Service Pack applied	Yes	
<b>Database</b>		
PURGECOMPLETEDJOBS	DAILY command job	
MOVEHISTORYDATA	DAILY command job	
DELETESTATUSMESSASGES	DAILY command job	
Increase DB connections ( <i>setdbparm</i> )	400	
Reduce idle connection time (seconds)	1200	
Oracle DB : Set Table partitioning for large tables		
Oracle DB : Change Initrans	20	
Oracle DB : Increase CLOB size (KB)	16	
MS-SQL: Pre-grow database tables		
MS-SQL: Provide separate devices for heavily used tables		
MS-SQL: Relocate logs and TEMPDB on a separate device		
<b>Agent</b>		
Install 64-bit agent on 64 bit OS	Yes	
Enable spool clean up in agenparm.txt:		
■ runnerplugin.spool.clean.enable=	■ True	
■ runnerplugin.spool.expire=	■ 10D	
■ runnerplugin.spool.sleep=	■ 24H	
Enable Disk space monitoring		
■ agent.resourcemon.enable=	■ True	
■ agent.resourcemon.threshold.disk.critical=	■ 50M	

Type	Recommended Settings	Enter Your Settings
<ul style="list-style-type: none"> <li>■ agent.resourcemon.threshold.disk.warning.notice=</li> <li>■ agent.resourcemon.threshold.disk.warning.severe=</li> </ul>	<ul style="list-style-type: none"> <li>■ 200M</li> <li>■ 100M</li> </ul>	
Latest Patch and Service Pack applied	Yes	
<b>Additional settings and parameters</b>		