

CA Workload Automation DE (dSeries Edition) r11.1 & r11.3 Cookbook



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 © 2013 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

TITLE AND PUBLICATION DATE:

CA Workload Automation DE Cookbook
Publication Date: June 26th, 2013
Last Update June 12th, 2015

Table of Contents

CA WORKLOAD AUTOMATION DE SOLUTION OVERVIEW	6
KEY FEATURES DIFFERENTIATING CA WA DE FROM OTHER SOLUTIONS	7
CA WA DE INSTALLATION.....	10
INITIAL INSTALLATION GUIDELINES	10
UPGRADE GUIDELINES	10
MULTI-TIERED INFRASTRUCTURE DIAGRAM	10
COMMUNICATION PORTS DIAGRAM.....	11
USING SQL EXPRESS 2008	12
STARTING AND STOPPING THE WORKLOAD AUTOMATION COMPONENTS	14
<i>Starting the Workload Automation Server</i>	<i>14</i>
<i>Stopping the Workload Automation Server</i>	<i>14</i>
<i>Starting Workload Automation Agents</i>	<i>14</i>
<i>Stopping Workload Automation Agents</i>	<i>15</i>
<i>Checking the Agent Status.....</i>	<i>15</i>
IMPLEMENTING A UNIX/LINUX INIT.D STARTUP SCRIPT ON R11.1.....	15
<i>Solaris init.d startup script example</i>	<i>15</i>
APPLYING PATCHES AND NEW BUILDS	15
CA WA DE SERVER	17
WORKLOAD AUTOMATION MAINTENANCE BEST PRACTICE	17
<i>Data Retention.....</i>	<i>17</i>
Workload Automation Server Log Files	17
System Agent Spool and Log Files.....	17
Server Audit Tracking and Audit Logs	17
<i>Backups.....</i>	<i>18</i>
Exporting/Importing Workload Automation Repository Artefacts.....	18
Bulk Migration of CA Workload Automation Artefacts to a New Repository	20
Exporting/Importing Desktop Client Custom Views and Preferences	20
<i>Restoring the Workload Automation Repository.....</i>	<i>21</i>
Important Notes Regarding Importing vs. Restoring the Repository.....	21
CONFIGURING CA WA DE TO USE ACTIVE DIRECTORY.....	21
<i>Importing LDAP users</i>	<i>24</i>
<i>Changing a user's Authentication mode.....</i>	<i>24</i>
CONFIGURING CA WA DE TO WORK WITH NIMSOFT	25
WORKLOAD AUTOMATION MIB.....	26
REPLACEMENT OF SERVER.LOG FILES WITH STDOUT.TXT FILES ON UNIX	26
VALIDATE DATABASE CONNECTION UTILITY	26
CHECK INTEGRITY OF SERVER PATCH FILES UTILITY	26
CLEANUP UTILITY.....	26
CHANGING THE SNMP DEFAULT PORT	27
SUPPORT FOR WEB SERVICES.....	27
LIMIT SIZE OF SPOOL FILE IN EMAIL ATTACHMENTS	27
CONFIGURE THE MAXIMUM NUMBER OF ACTIVE APPLICATION GENERATIONS	27
WORKLOAD AUTOMATION AUDIT LOG SCENARIO	28
DISASTER RECOVERY (DR) PROCESS FOR THE CA WA DE SERVER.....	34

CA Workload Automation DE (dSeries Edition) Cookbook

Steps to switch the Sandbox server to the DR Role.....	34
Returning to normal operations	36
Steps to switch back to the Sandbox Role.....	37
CA WA SYSTEM AGENTS AND PLUG-INS	38
MODIFYING LINUX FIREWALL TO CATER FOR WA SYSTEM AGENTS	38
PREVENTING JOBS FROM RUNNING UNDER A “ROOT” USER ID	38
CONFIGURING SYSTEM AGENT USING THE “SILENT INSTALL” CONFIGURATION FILE.....	39
QUICK SELF-CHECK FOR DE SERVER AND AGENT CONNECTIVITY ISSUES.....	43
<i>Check server parameters</i>	44
USING CA WADE WITH OPENBSD/FREEBSD.....	45
CA WA DE DESKTOP CLIENT	47
DISABLE BROWSER DEBUGGING FOR ONLINE HELP	47
CA WA DESKTOP CLIENT WORKSPACE.....	47
SECURITY PERMISSIONS	47
GETTING HELP WHEN USING THE DESKTOP CLIENT	47
PLANNING YOUR SITE’S NAMING CONVENTIONS	47
<i>Server Names</i>	48
<i>Agent Names</i>	48
<i>Application Names</i>	48
<i>Job Names</i>	48
<i>Event Naming</i>	48
<i>Calendar Names</i>	49
<i>WORKDAYS</i>	49
<i>Holidays</i>	49
<i>Special Days</i>	50
<i>Alerts</i>	50
<i>Resource Names</i>	50
<i>JavaScript Names</i>	51
ROLE-BASED SECURITY	51
<i>Security</i>	51
<i>User and Group Clean-up</i>	51
<i>EVERYONE Group</i>	52
<i>Group Architecture</i>	53
<i>SYSADMINS Group</i>	53
<i>TRAINING Group</i>	53
DEFINING WORKLOADS	54
SINGLE BACKSLASH AND DOUBLE QUOTES NOT RECOGNIZED.....	54
EDIT JOB DEPENDENCIES FEATURE	54
SCHEDULING BASED ON TIME ZONES.....	54
SCHEDULING AN EVENT ON A PARTICULAR DAY	54
SCHEDULING TERMS.....	55
HOW TO USE VIRTUAL RESOURCES.....	58
WHAT DO THOSE APPLICATION PROPERTIES CONTROL?.....	59
<i>Wait for previous generation</i>	59
<i>Propagate dueout time</i>	59

CA Workload Automation DE (dSeries Edition) Cookbook

<i>Require reason for job commands</i>	60
<i>Do not trigger if active</i>	60
TAKING ALTERNATIVE PATHS THROUGH THE JOB STREAM	60
SUBAPPLICATION: ORCHESTRATING PARALLEL RUNNING OF APPLICATIONS	62
SUBAPPLICATION: GROUPING JOBS	64
MANAGING/MONITORING WORKLOADS BY EXCEPTION	65
<i>Creating your own personalised Custom View</i>	65
RUNNING A WINDOWS COMMAND PROMPT COMMAND	67
RUNNING A JOB TO INTERACT WITH A WINDOWS APPLICATION	68
RUNNING AN APPLICATION EVERY “N” MINUTES OR HOURS	71
SKIPPING NON-ESSENTIAL WORKLOADS DURING DISASTER DAYS.....	72
USING FTP WORKLOAD OBJECTS	73
<i>Setting up an FTP User in CA WA DE</i>	73
<i>Defining an FTP workload object</i>	74
DELAYING JOB EXECUTION BY LESS THAN A MINUTE.....	75
USING THE DESKTOP CLIENT’S SNMP MESSAGES VIEWER	75
<i>Define the structure of your SNMP message</i>	75
<i>Setup a temporary SNMP manager on your workstation</i>	76
<i>Run jobs to generate SNMP messages</i>	77
<i>Cleanup</i>	78
INSERTING CVS SUBSTITUTIONAL KEYWORDS INTO CA WA DE APPLICATIONS.....	78
USING GPG (OR PGP) WITH SYSTEM AGENTS	80
JAVASCRIPT IS NOT JUST FOR WEB PAGES.....	84
<i>JavaScript Introduction</i>	84
<i>Java vs. JavaScript</i>	84
<i>Creating and Running a Simple Script</i>	84
<i>Basic JavaScript in an HTML Document</i>	85
<i>Some General Notes</i>	85
<i>Basic Elements of JavaScript</i>	85
<i>JavaScript Summary</i>	89
GETTING SUPPORT FOR CA WA DE	90
WHAT YOU GET	90
SEVERITY 1 CASES.....	90
TELEPHONE SUPPORT	90
ONLINE SUPPORT	91
GETTING REGISTERED TO CA SUPPORT ONLINE	91
LOGGING IN TO CA SUPPORT	92
UPDATING YOUR SUPPORT PROFILE/PASSWORD	93
CA WA DE PRODUCT HOME PAGE	93
NAVIGATION PANE	95

CA Workload Automation DE (dSeries Edition) Cookbook

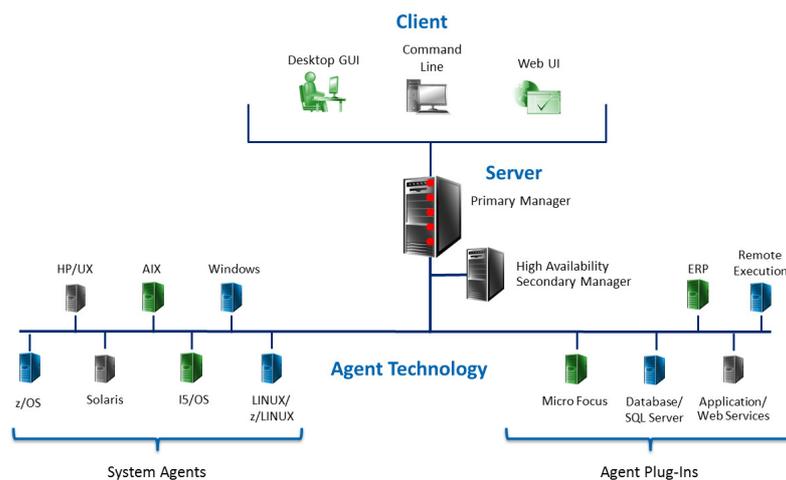
ACKNOWLEDGEMENTS

CA Workload Automation DE (dSeries Edition) is a very flexible solution and can take a number of years to understand and become proficient with all its capabilities. It is therefore not uncommon to discover a number of different ways to solve a business problem in CA WA DE. The best rule of thumb is to use the easier to understand and most maintainable method. The material in this Cookbook has come from many sources to help provide insight and guidance; such as from the CA support site at <http://support.ca.com>, our implementation partners, as well as from CA technical staff. Should you wish to make comment on the content or contribute additional material to this document, please email the Cookbook's editor at alan.monument@contractor.ca.com or monument@bigpond.net.au.

CA Workload Automation DE Solution Overview

The functionally rich multi-tiered architecture of the CA Workload Automation DE (dSeries Edition) solution aims to eliminate the complexity and cost associated with the management of multiple and differing job scheduling solutions. It assists with reducing the amount of manual intervention associated with workload management and job scheduling through some key differentiating capabilities.

The terminology, basic components, and architecture of the CA Workload Automation DE Framework are as follows:



CA Workload Automation DE Server [the “Manager”]: A CA Workload Automation DE Server runs on a distributed platform and serves as the central point of control for all workload on all the other platforms/servers in the enterprise. CA’s Workload Automation Server is aware of the global environment, make decisions and issue commands. It serves as the decision-maker, determining when a job is eligible to run based on predecessor requirements, available logical resources, date/time, and feedback from System Agents, file or database activities, etc. The CA Workload Automation Server is responsible for orchestrating all of the workload through the Agent Technology and as such is able to monitor SLAs and proactively raise alerts when business processes start to go wrong. An optional High-Availability Server provides seamless failover and fall back capabilities should the primary scheduling server fail for any reason.

System Agents: The CA Workload Automation *System Agent* is an intentionally small program that resides on any server platform (real or virtualised) that needs to orchestrate business process workloads. The agent is aware of the local environment, acts on commands received by the *CA Workload Automation Server* and

CA Workload Automation DE (dSeries Edition) Cookbook

forwards status information back to the Server. A wide range of CA dSeries Agents exist covering most commercially available computing platforms and a rich set of *Agent Plug-Ins* extends the agents functionality.

CA Workload Automation DE Desktop Client GUI: The CA Workload Automation Desktop Client GUI is an Eclipse-based interface. As the primary user interface, it is used to define, monitor, and manage the organisation's workloads. A Command Line interface allows commands to be invoked from command lines or scripts, and a Web User Interface allows monitoring of workloads from a web browser.

All components within this Automation Framework communicate through TCP/IP. These "plain text" messages are encrypted for security purposes. At installation time, each site selects a unique encryption key to be used for the encryption of all these messages. Agents listen on a TCP/IP port waiting for instructions to perform some type of work. Once a job starts executing, the agent formats and transmits a message back to the CA Workload Automation Server so that the job's status can be updated. It uses the same encryption method, via TCP/IP, that sent the message to the agent in the first place.

Workload Objects: Workload Objects are a representation of the managed work, such as the execution of a program or a script, the detection of a row added to a database, the presence/absence of specific information in a file, the invocation of a session bean, the expansion of a file by a certain percentage, and many other conditions.

Key features differentiating CA WA DE from other solutions

Schedule Definition from a single point – We believe CA's Workload Automation is *unique* in that scheduling and development staff can define within a *single Application* definition an *entire business process*, end-to-end, regardless of run frequency, job dependencies, or operating platform; providing a single management perspective. In addition to the clarity and understanding this generates across the business and IT divisions, it is a major efficiency factor in the creation and ongoing maintenance of schedule definitions, and is further enhanced by the ability to use the single definition for multiple (hourly, daily, weekly, monthly, on demand, etc.) invocations of an application. The use of variables within each Applications job definitions can further reduce the number of job definitions that need to be created and maintained.

Dependency Inheritance – CA Workload Automation's ability to allow the dependency from one job to another to be inherited by a successor when an intervening job is not scheduled to run is one of the unique capabilities that enables a single definition for all iterations of a schedule to be created. This means there will no longer be the need to define separate definitions for daily, weekly, monthly and/or end-of-quarter runs for a particular schedule – *this can save countless person-hours in the creation and maintenance of schedule definitions, and reduces errors in keeping changes synchronised.*

No Schedule Load – The concept of the 'daily schedule load' or a 'New Day process' does not exist within CA Workload Automation. In fact individual schedules can run or be held across multiple days *without any special effort or consideration*, and multiple iterations of an application can run many times within the same day without requiring multiple schedule definitions. With more of an organisation's workload being real-time, it is important that work can be initiated whenever it is required to be run. CA Workload Automation automatically schedules each *Application* individually, according to the actual business needs, not an enforced daily schedule; and multiple iterations of an application can be either be run in parallel or queued to run one after the other, simply by ticking an option box.

Integration for End-to-End Automation – CA's Workload Automation Agent technology enables the management of processes across hundreds or even thousands of individual disparate servers within an organisation's IT infrastructure, from a single point, without introducing overhead or reducing performance.

CA Workload Automation DE (dSeries Edition) Cookbook

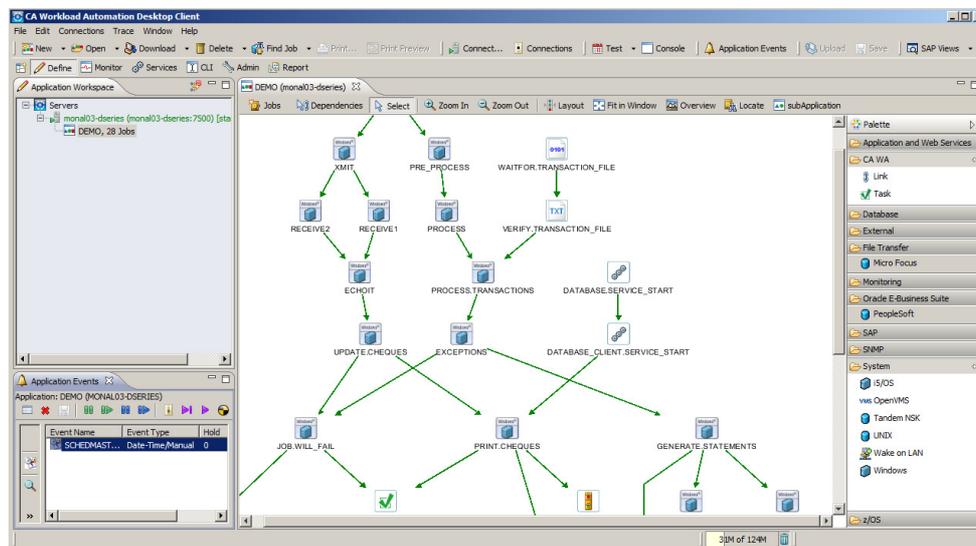
CA Workload Automation Agents are *lean, non-invasive and powerful* event-driven applications that act as conduits to extend and optimise business process automation. No additional tools, products, or servers are required to provide users of the solution with an enterprise view of their workload. The agent technology has a rich set of built-in workload automation monitoring capabilities that provide additional information to enhance the decision making capabilities on where best to place and when to run business processes.

Simple Calendar Management – CA Workload Automation’s inherent understanding of common scheduling terms provides a flexible English-like language for scheduling workload. A calendar is simply a “dictionary” of scheduling terms used as a repository for Holiday and Special Day definitions resulting in a minimum number of calendars being required. Users can define their organisation-specific scheduling criteria to a calendar to cater for holidays or special processing days such as “payday”, “billing_day”, “statement_day”, or “end_of_quarter”. These calendar entries can be defined many years in advance.

Real-time Simulations – With CA Workload Automation users are able to simulate job flows prior to production implementation to assure an *error-free schedule definition*. Users will be able to simulate any schedule, in the past or future, at any time. In addition, symbolic variables are resolved in the simulation allowing users to see the parameter variations with different simulations. This not only provides the organisation’s operations personnel with the comfort of knowing how the schedules will run, but will give the application development staff the comfort in knowing that the schedules are defined correctly. This is a great asset in the application development life-cycle to aid in quality assurance by allowing users to validate their expected results. Simulation can prevent production errors and is a vital capability to ensuring a successful conversion from another scheduling solution.

Service level management across all services – CA’s solution provides exceptionally strong service level management capabilities. Our technology allows the business to establish SLAs with either explicit or implicit times of completion for the work requested and facilitates meeting or exceeding the service levels established by the various business units for the workloads being managed.

Graphical User Interface for Workload Management – CA Workload Automation Desktop Client software provides a powerful graphical user interface for centrally defining, monitoring and controlling the organisation’s entire enterprise workload. It is a significant productivity tool both for those defining and scheduling applications, and for those who monitor and control them. It’s an easy intuitive graphical interface enabling even the newest users to quickly drag-and-drop their way through workload definitions, calendar management, and to the monitoring and control of the enterprise workload.



CA Workload Automation DE (dSeries Edition) Cookbook

The Desktop Client promotes more efficient use of IT resources by enabling the organisation's staff to clearly and simply communicate their application's scheduling requirements. A flow-charting tool enables application developers to detail workflow components, platforms, and dependencies. To complete this definition of an *Application*, all that is then needed is some additional information about each workflow component; what needs to execute (the program/command), where it needs to run (which server or group of servers), and when it should run – no specific platform skills are required by staff to schedule this work.

This clear presentation of technical information greatly reduces errors and misunderstandings associated with application scheduling during hand-over from application development to operations.

CA WA DE Installation

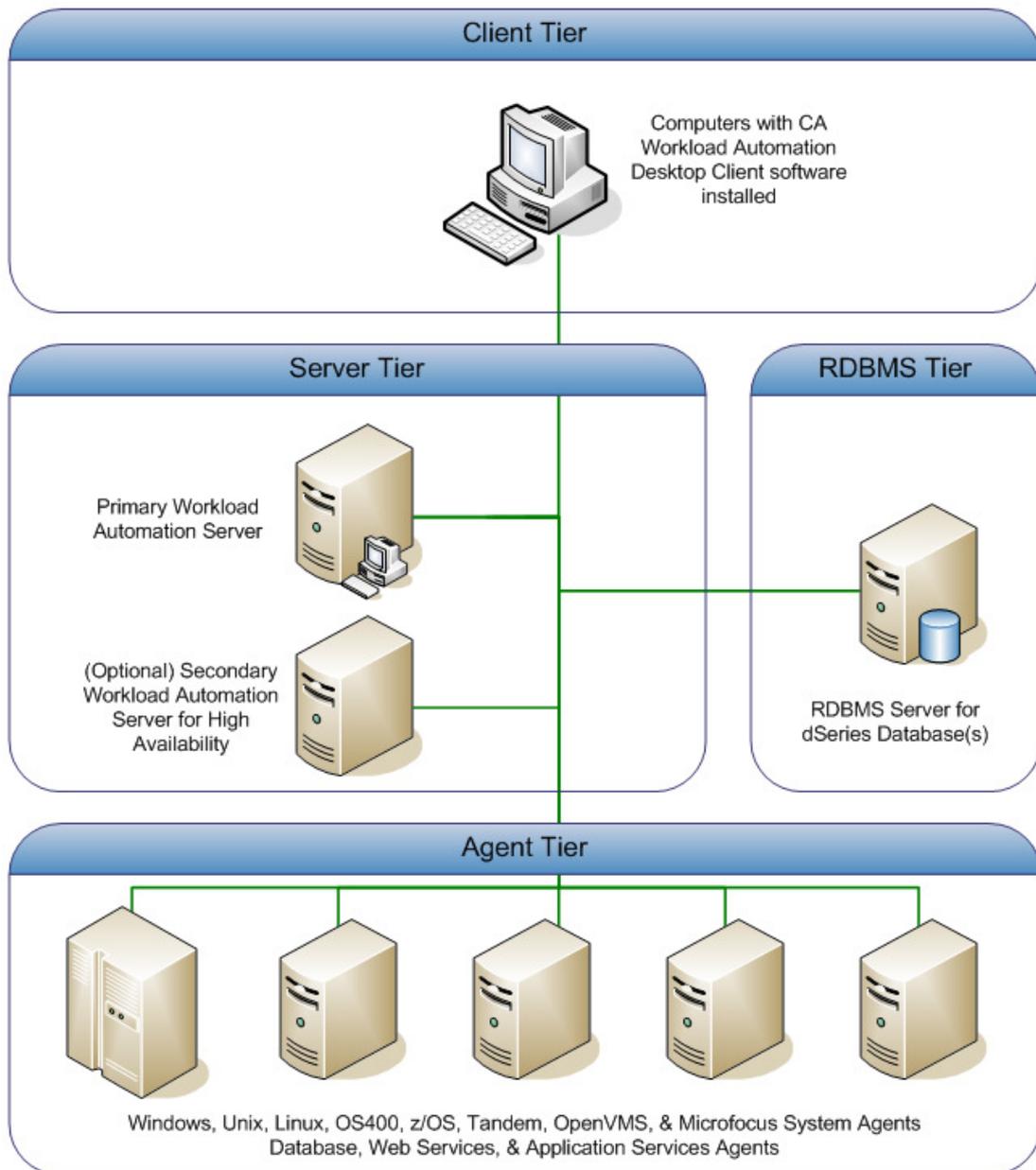
Initial Installation Guidelines

To be written

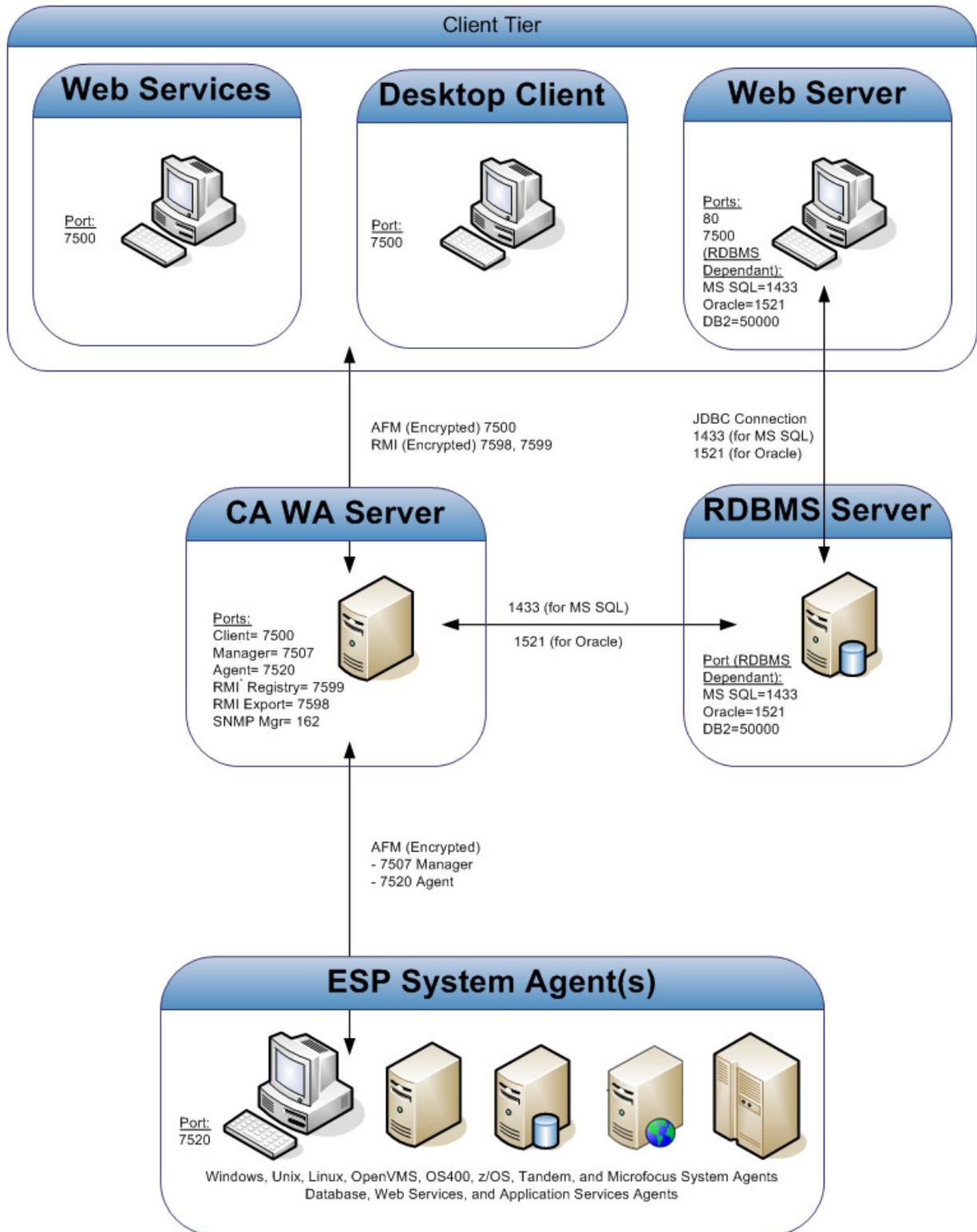
Upgrade Guidelines

To be written

Multi-Tiered Infrastructure Diagram



Communication Ports Diagram



RMI = Remote Method Invocation. See <http://java.sun.com/developer/onlineTraining/rmi/RMI.html> for details and firewall implications
 AFM = dSeries TCP/IP protocol used to communicate between the server (manager) and remote components (agents, clients)

Using SQL Express 2008

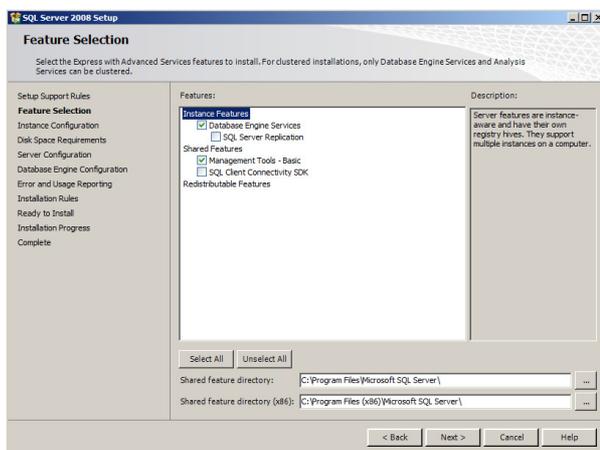
Contributed by Alan Monument

Although not officially supported by CA, it is possible to build “sand pit” instances of CA WADE server using SQL Express 2008. This is very useful for creating short-lived instances of CA WADE to test out some desired scheduling situations without needing to impact the production environments. The advantage of using SQL Express 2008 is that it does not require a license to run in non-production environments. Please however discuss your CA WADE licensing requirements for such situations with your CA Representative to ensure you are operating within your contractual obligations.

Download the SQL Express 2008 media from a Microsoft web site. Ensure that you download the “with Tools” version of the media as that includes the management tools you will later need to create the CA WADE database (i.e. http://download.microsoft.com/download/7/9/4/794bfafa-aea7-45d4-a6ea-4e92f09918e3/SQLEXPRTW_x64_ENU.exe).

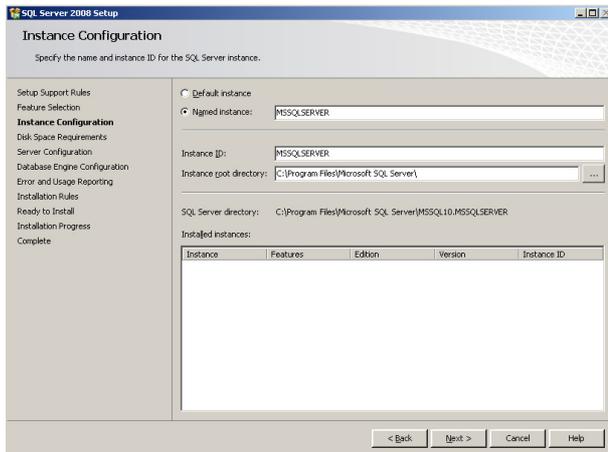
Use *Run as Administrator* to run the SQL Express 2008 installer program. During this installation you will need to observe the following so that the Express version of MS SQL 2008 will work properly with CA WADE:

- When installing SQL Express 2008 on Windows Server 2008, you may be prompted to “Use the Role Management Tool to install .NET Framework 3.5 on the Server”. If you get this message:
 - Access the Role Management Tool by clicking on **Start > Admin Tools > Server Manager**
 - Click on the **Features** item in the left pane of the Server Manager application and then click on the **Add Features** hyperlink in the right of the Features Summary window.
 - In the resultant *Add Features Wizard* dialog window, select (tick) the **.NET Framework 3.5.1 Features** item. If you get a dialog indicating that other role services also need to be installed, simply click on the **Add Required Role Services** button to continue.
 - Click on the **Next** button on the next 3 windows, followed by the **Install** button on the *Confirmation* window, and then finally the **Close** button to complete the installation.
 - Rerun the SQL Express 2008 installer program.
- When installing ensure that at least the **Database Engine Services** and **Management Tools - Basic** features are selected in the *Feature Selection* step...

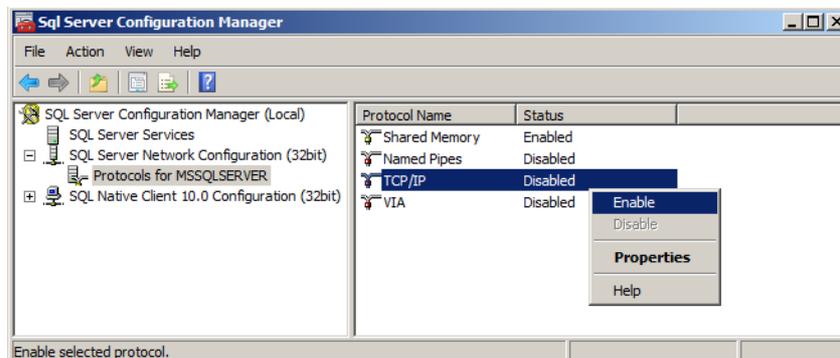


CA Workload Automation DE (dSeries Edition) Cookbook

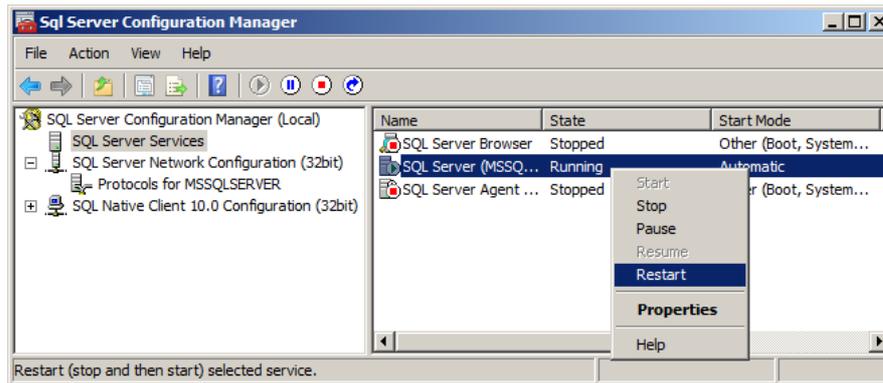
- In the *Instance Configuration* step, **DO NOT** use the *default instance* setting. Select the *Named instance* radio button and replace the defaulted name with **MSSQLSERVER** instead...



- Note that if you are installing SQL 2008 Express onto a Windows 2003 Server you might encounter an exception during the installation process as it will not be able to start the SQL Agent. Don't be concerned by this error and let the installation process continue. The encountered exception will be listed at the end of the installation process – again don't worry too much about this as it doesn't affect the operation of CA WADE (as far as I can tell).
- The last thing to do is to ensure the SQL Protocols that CA WADE needs to use are enabled for your MSSQLSERVER instance - TCP/IP is disabled by default, inhibiting its use by CA WADE. Enable this protocol *before* attempting to install CA WADE:
 - Click the **Start** button -> **Microsoft SQL Server 2008** -> **Configuration Tools** -> **SQL Server Configuration Manager...**



- Expand out the *SQL Server Network Configuration* object, and then click on the revealed *Protocols for MSSQLSERVER* object to see a list of the protocols used. Right mouse click on the **TCP/IP** entry and **Enable** that protocol.
- Click on *SQL Server Services* object in the left hand navigation pane, right mouse click on the *SQL Server (MSSQLSERVER)* service entry in the right pane and select **Restart** to restart the service and effect the above change.



- Finally don't forget to apply the latest MS SQL Server 2008 Service Pack. SP3 was the most current at the time of writing (<http://www.microsoft.com/en-us/download/details.aspx?id=27594>).

You will then be ready to install CA WADE.

Starting and Stopping the Workload Automation Components

A complete description for starting and stopping Workload Automation components can be found in the *Workload Automation Implementation Guide* and *System Agent Administrator's Guide*. The following is provided as a quick reference for starting/stopping Workload Automation components from the command line.

Starting the Workload Automation Server

- Change to the directory where the server is installed.
- Enter the following command:

```
./startServer
```

A message is displayed indicating that the server started.

Alternatively, you can enter **./startESP** to start the server and the default CA WA System Agent at the same time.

Stopping the Workload Automation Server

- Change to the directory where the server is installed.
- Enter the following command:

```
./stopServer
```

A message is displayed indicating that the server stopped.

Alternatively, you can enter **./stopESP** to stop the server and the default

Starting Workload Automation Agents

- Change to the directory where the System Agent is installed.
- Enter the following command:

```
./cybAgent &
```

The ESP Agent runs in the background.

Stopping Workload Automation Agents

- Change to the directory ESP Agent is installed in.
- Enter the following command:

```
./cybAgent -s
```

Checking the Agent Status

The `status.os` file provides accurate information on the status of the System Agent. From the directory the System Agent is installed in, enter the following command:

```
cat status.os
```

Implementing a Unix/Linux `init.d` startup script on r11.1

When installing the management server and agents on a Windows server, the windows services can be flagged to optionally start automatically, such that when a server is restarted, the manager and/or agent services will also automatically start. On Unix/Linux based systems the `init.d` startup script will need to be modified to automatically start these services. The following is an example of how this can be accomplished using a Solaris based Unix system as an example.

Solaris `init.d` startup script example

Contributed by Ron Harrison

espinit - is installed on the management server and is used to start both the CA WA DE server and local agent

espagent - is installed on systems where-ever there is a system agent

To install the CA WA DE `espinit` server script:

- copy to `/etc/init.d/espinit`
- then link `/etc/init.d/espinit` to `/etc/rc3.d/S999espinit` and to `/etc/rc1.d/K000espinit`:
 - `ln -s /etc/init.d/espinit /etc/rc3.d/S999espinit`
 - `ln -s /etc/init.d/espinit /etc/rc1.d/K000espinit`

Repeat the above steps using the `espagent` script on those servers where system agents are deployed.

Applying Patches and New Builds

This section covers what you might need to do to apply any updates to the CA Workload Automation DE components.

The basic rule of thumb when applying updates to CA Workload Automation DE components is to apply the changes in the following sequence to ensure that the components are able to continue to communicate with each other:

1. Desktop Client
2. CA Workload Automation Server
3. Agents
4. Any other component

The basic update process is:

- 1) the build is extracted to the Workload Automation Server's <installDir>/update directory
- 2) the Workload Automation Server is stopped
- 3) the changes are applied by running an *update* script
- 4) the Workload Automation Server is restarted.

The patch media contains instructions to perform this process.

If you obtain a newer build of the **Desktop Client** from CA Support, it will be in the form of a replacement *setup.exe* file which you can simply run to install the newer version (after de-installing the earlier version). Any settings used in the previously installed version will be retained in the newer version.

CA WA DE Server

Workload Automation Maintenance Best Practice

Data Retention

Workload Automation Server Log Files

The CA Workload Automation Server maintains the following log files to store diagnostic and auditing information about the Server:

- **Trace log**
Stores communication messages between the Workload Automation Server and components, and maintains debugging information. Its filename is *tracelog.txt*.
- **Automated Framework Message (AFM) log**
Stores communication messages between the server, agents, Desktop Client, and WA Web Client. Its filename is *afmlog.txt*.
- **Audit log**
Records user input. The server creates a new audit log every day. You cannot change this frequency.

These log files accumulate over time and use. To maintain performance, CA recommends that these files are cleared regularly. The server also logs information about completed jobs in the completed jobs repository in its database, which can also be cleared.

System Agent Spool and Log Files

The Workload Automation System Agent software creates **spool files** during its normal operation. By default, these spool files **do not clear automatically**, and they need to be cleared periodically. If the file system where a spool file resides reaches its maximum size, the System Agent cannot continue to run.

Note: The System Agent does not limit the size of the spool files. Spool files are limited in size by the available space on the file system of the system on which they reside.

The System Agent software also keeps a set of **log files** that must be cleared periodically to maintain disk space availability. The log files contain records of all messages between System Agent and the Workload Automation server, as well as internal messages. These files are located in the log directory by default and are updated continually while the System Agent is running. The types and number of logs that are generated depend on the *log.level* parameter set in the *agentparm.txt* file.

Server Audit Tracking and Audit Logs

Workload Automation Server's Audit Log file stores information about user activities and their input while using the solution which is useful for Sarbaines-Oxley compliance. This information can be exported at anytime, for any given date range, by running a command called *exportauditlog*.

For example the command `exportauditlog path("/WA_AuditLog") name("20131007_Log") startdate("20131007") enddate("20131007")` would export the audit log into 2 files in a new directory called 20131007_Log under the /WA_AuditLog directory, containing audit file entries for the date of 7th October 2013. This command may be

CA Workload Automation DE (dSeries Edition) Cookbook

constructed from system variables and run from a scheduled job, or from the Command Line Interface (CLI) in the Desktop Client.

This export command generates files in both *comma seperated values (CSV)* and *HTML* formats. For each function performed by Workload Automation there are 2 records generated in the Audit Log; a **request** and a **response**. Each of these records is divided into columns:

1. A **Date & Time Stamp** (i.e 20130821 19:48:57.563)
2. A **Correlation ID** (this is a system generated value and is the same in both the request and the response records)
3. The **Request / Response** identifier (req or resp)
4. The **User ID** of the person performing the function
5. The **Command Type** (application, action, event,...)
6. The **Command Name** (action release for example to release a job)
7. **Command / Results** (details about the object in question in the request record, and the results of the function in the response record)

Backups

To back up CA Workload Automation DE there are two separate components that need to be backed up; the **Workload Automation Server** software and the **Workload Automation Repository database**.

Use your normal backup processes to back up the directories in the **Workload Automation Server's** installation directory. With the exception of the two directories highlighted below, which may require a more regular backup regime, most of the rest of the directories and associated files under this installation directory will remain fairly static over time:

- configuration and license files (`//CA/Workload Automation/Resources`)
- log files (`//CA/Workload Automation/LogFiles`)

The **Workload Automation Repository database** is the solution's most volatile data store and requires a regular database backup regime to enable recovery of frequently changing artefacts in the case of an accidental deletion or incorrect modification of an object. Workload Automation *Application* definitions are the most regularly changed objects and the solution's built-in versioning capability allows developers to optionally recover to an earlier revision of a specific definition when they *Download* an Application from the Workload Automation repository using the Desktop Client.

Important Note: the Workload Automation Repository database stores server specific configuration information; therefore it is important to restore the repository to the correct environment as it will only work with its original "owning" Workload Automation server.

Exporting/Importing Workload Automation Repository Artefacts

The best way to back up a Workload Automation Repository is to use the *imexutil.sh* import export utility (see the *CA Workload Automation Implementation Guide* for details). *imexutil* is used to export Workload Automation Repository objects into XML source files and provides a granular way of managing individual artefacts.

CA Workload Automation DE (dSeries Edition) Cookbook

The following *imexutil* commands when run in Batch mode will export all artefacts from a CA Workload Automation DE Repository to individual directories under a mount point called `//WAExport`.

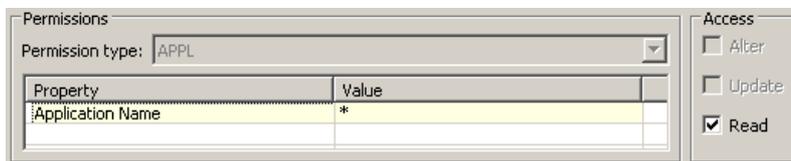
```
user admin admin
exportagent -outdir //WAExport/Agents
exportagentgrp -outdir //WAExport/AgentGrps
exportalert -outdir //WAExport/Alerts
exportapplication -outdir //WAExport/Applications
exportcalendar -outdir //WAExport/Calendars
exportconfig -outdir //WAExport/Configs
exportevent -outdir //WAExport/Events
exportforecast -outdir //WAExport/Forecasts
exportgroup -outdir //WAExport/Groups
exportjavascript -outdir //WAExport/Javascripts
exportresource -outdir //WAExport/Resources
exportsundry -outdir //WAExport/Sundry
exportuser -outdir //WAExport/Users
exportvarctx -outdir //WAExport/Variables
```

These commands can be placed in a file (i.e. `ExportAll.txt`) which is then referenced in the Arguments passed to an `imexutil.sh` command, as in the following example:

```
-h esp.prd.com.au -p 7500 -c @"//WAExport/ExportAll.txt"
```

Important note: Newer releases of CA WA DE may add additional object types to the repository. In this case appropriate export commands will need to be added to the above commands to ensure that these new objects are also exported as part of your regular backup regime.

Use an appropriate user in the above *user* command which has read access rights to all applications in the repository. Alternatively, ensure the admin user has appropriate access rights by adding the APPL permission type as highlighted below...



These XML files can be imported at any time back into the Workload Automation Repository or into another Repository; and can optionally be checked into a software version control solution, such as CA Change Manager, for greater Production rollout control and audibility.

These exported XML files should be backed up to external media on a regular basis as per corporate backup and archive policies.

A complete restore of a Workload Automation Repository or the population of a new repository can be accomplished simply by importing all (or some of) the previously exported XML files.

Additionally, Desktop Client users can also create local versions of XML source files of the Application definitions they are working on by issuing a *File/Save As...* menu command to create the XML file on their computer's file system, and can then later read that XML file back in and open it in the Desktop Client's Define perspective's Application Workspace by using the *File/Open* command. It is therefore good practice for developers to use this capability to save copies of these files locally; which will provide them with an insurance policy against potential accidental deletions, or to enable them to perform "offline" changes while disconnected from the corporate network and the CA WA DE Repository.

Bulk Migration of CA Workload Automation Artefacts to a New Repository

The following procedure describes how it is possible to do a bulk Migration of CA Workload Automation Artefacts to a New Repository and was taken from Solution Document RI21926 available from the CA Support site. Before using this procedure please refer to the most recent version of the document, which may include updates.

Note that the *imexutil* import export utility is not designed to import security artefacts. This is by design and the following procedure is therefore the *only* method of migrating exported Workload Automation *users* and *groups* between repositories – also see Knowledge Base Article TEC523897 on the CA support site for further details.

1. Install a new CA Workload Automation Server on the destination environment where you want the data to be migrated.
2. Export the artifacts from existing CA Workload Automation Server:
Using the import export utility, export out all the artifacts (Artifact definitions include Alerts, Applications, calendars, Events, Forecasts, JavaScript scripts, Resources, Users, GlobalVariables and Groups) from the CA Workload Automation server into respective folders.
3. Modification of **DBInit** (this folder is located in <installdir>/Resources) on new CA Workload Automation server:
 - a. Make a backup of the DBInit folder.
 - b. Remove all folders in DBInit except Config.
 - c. Create folders for each of the artifact types: "Alert", "Group", "Javascript", "User", "ResourceDefinitions" under "Resource" folder under DBInit folder. Copy the exported artifacts from above step into these respective directories.
 - d. Move all the exported files from their respective folders into the new CA Workload Automation Server.
 - e. Ensure you rename the entire exported files with the **.xml** extension in the above created folders.
4. Populating the artifacts:
To ensure that the DBInit folder is read by CA WA during startup, you need to create an empty file and name it **populate**. Create this empty file in the <installdir>/Resources/DBInit folder.
5. Import Global Variables (if required)
The 'importvarctx' command in 'imexutil' lets you import global variable contexts which were earlier exported.
 1. Restart the New CA Workload Automation Server and check the tracelog to ensure that the log has "populate" messages.
 2. Log on and check if all the data have been successfully migrated or not.

Exporting/Importing Desktop Client Custom Views and Preferences

Custom Views created in the Monitor perspective and any Preferences established on each user's Desktop Client can be exported from and later selectively imported back into the Desktop Client. This allows these two user-customizable definitions to be easily restored after a desktop rebuild, copied to a new computer, or share the definitions with other users.

Custom Views and Preferences can be exported and Imported from options under the File menu option in the Desktop Client. It is recommended that all users are made aware of this capability and the sharing of Custom Views encouraged for operational consistency.

Restoring the Workload Automation Repository

Important Notes Regarding Importing vs. Restoring the Repository

When Events are exported, the *Next Scheduled Time* will also be exported into the resultant XML files as a STARTING time. However, when the XML is imported back into the Workload Automation Repository, the scheduling criteria for the Event is used to calculate the Next Scheduled Time occurring **after** the current (import) time. It will also restore any Hold or Suspend Counts associated with the Event. It is important to review all imported Events to ensure they reflect the desired state.

Alternatively, restoring a Repository from a database backup would bring in the *Next Scheduled Times* of all Events **as they were at the time the backup was taken**. Doing a normal “Warm Start” of the Workload Automation server after restoring its Repository will therefore run all Events from the restored *Next Scheduled Times* PLUS *all the intermediate scheduled times up to the present time*. If this is not what you intend to happen, it will be necessary to do a Cold Start of the server instead of the usual default Warm Start.

Please refer to the *Cold and Warm Start* section of the *Workload Automation Implementation Guide* for details of the types of, and the effects of running Cold and Warm Starts.

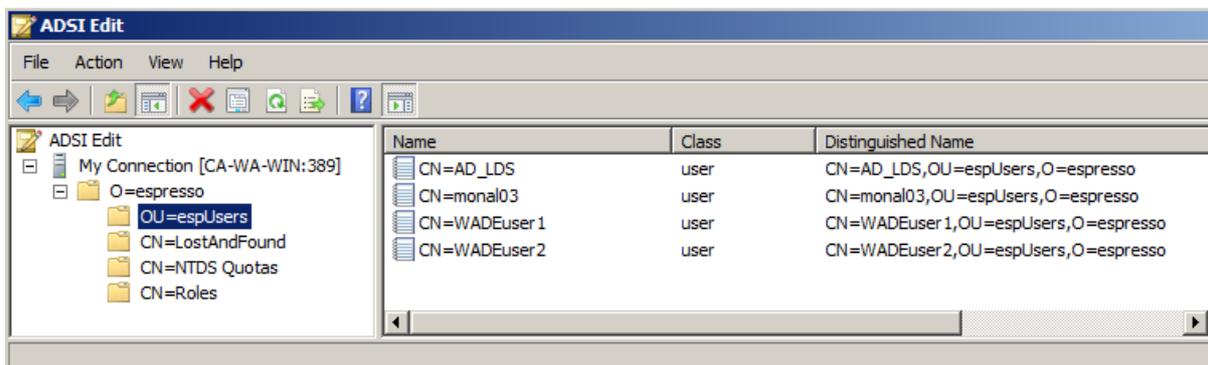
Important Note from the CA Workload Automation Implementation Guide! When you perform a normal cold start, the server deletes all active workload and associated states, and you lose all active processing data.

Configuring CA WA DE to use Active Directory

Contributed by Alan Monument

To enable user authentication via LDAP you will need to connect to your LDAP servers via the Desktop Client’s Topology browser (click on the *Admin* perspective button and then select the *Topology* tab in the left hand pane). This example configuration is using an instance of Active Directory Lightweight Directory Services (AD LDS) running on a Windows 2008 Server.

Below is an Active Directory Services Interface screen shot example showing some users defined in the AD LDS instance, with subdirectories used to define organisational units...



In the above example, user AS_LDS has been assigned to the *Administrators* role as this user will be the user that will be used to authenticate with LDAP, and all the other users have been assigned to the *Users* role.

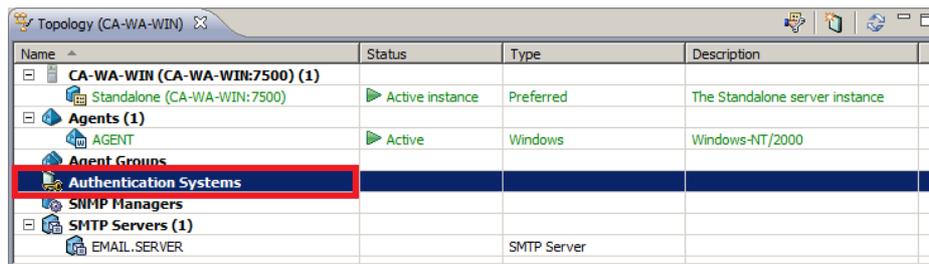
Note: By default, CA WA DE will not process any subdirectories when retrieving LDAP directory users. This behaviour can be changed by editing the `/conf/server.properties` file and uncommenting the `ldap.pullUsersFromSubdirectories` entry, and then changing the setting from `false` to `true`...

CA Workload Automation DE (dSeries Edition) Cookbook

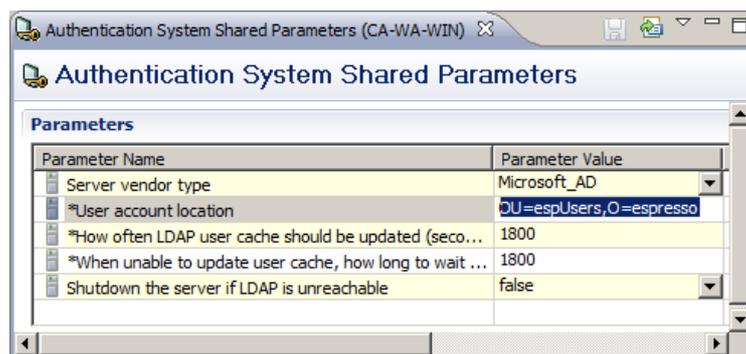
```
## Whether to retrieve ldap-users from 'configured ldap directory location' only or from its subdirectories as well?  
## Default (false): retrieves users from 'configured ldap directory location' only  
## Uncomment next line, and specify true if you want the ldap-users to be pulled from subdirectories as well.  
ldap.pullUsersFromSubdirectories=true
```

Important note: Always restart the CA WA DE server after making any changes to the above *server.properties* file to put these changes into effect.

To configure the Authentication system, either double click on, or right mouse click on the **Authentication Systems** item in the Topology pane and select **Configure Shared Authentication System** from the pop up menu...



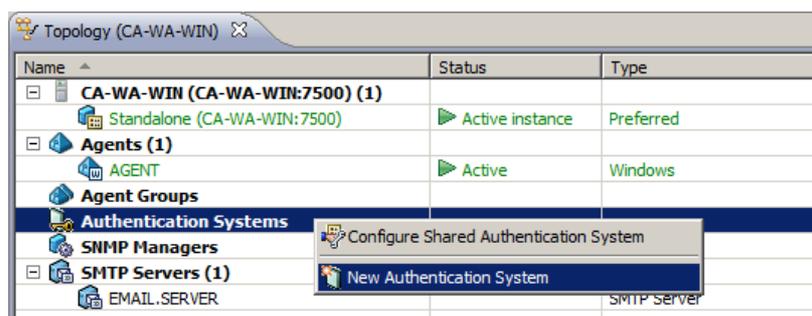
In the resultant *Authentication System Shared Parameters* pane, specify the LDAP server vendor type and modify the appropriate parameters...



User account location is the LDAP organisation location (see earlier *Active Directory Services Interface* screen shot) that contains the CA WA DE users.

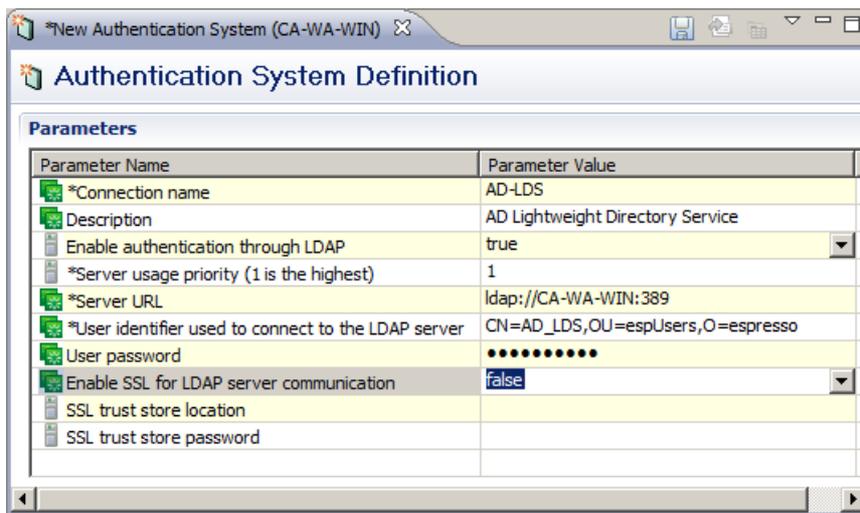
Important Note: If you only want to use LDAP to authenticate your users, then set **Shutdown the server if LDAP is unreachable** to *True* – this will then force a shutdown the CA WA DE server if all the LDAP servers are unreachable.

Save any changes you have made; then right mouse click the **Authentication Systems** entry in the Topology browser and then select the **New Authentication System** item from the pop up menu...



CA Workload Automation DE (dSeries Edition) Cookbook

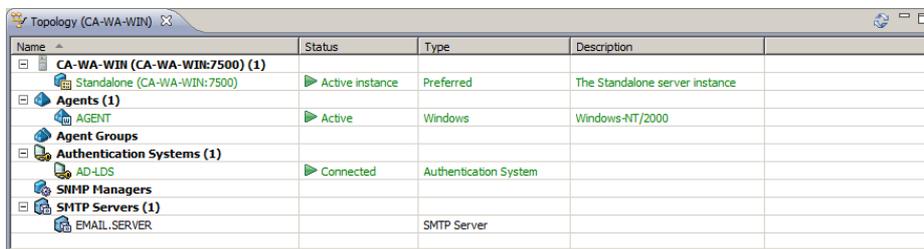
You will be presented with a new pane to define the LDAP system parameters.



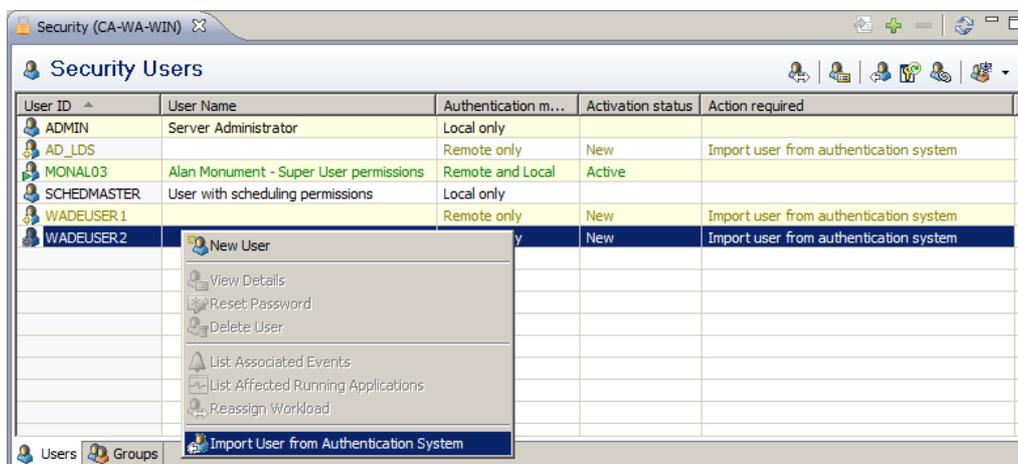
Note: Multiple LDAP servers can be defined and the **Server usage priority** determines the order in which the servers will be used (1 for the first, 2 for the second, etc.).

Save any changes. Once you have added all the LDAP servers in the Desktop Client it will be necessary to restart the CA WA DE server to enable it to connect to the newly defined LDAP server(s).

After the CA WA DE server has been restarted, use the Desktop Client's Topology browser again to ensure the LDAP server has connected successfully...

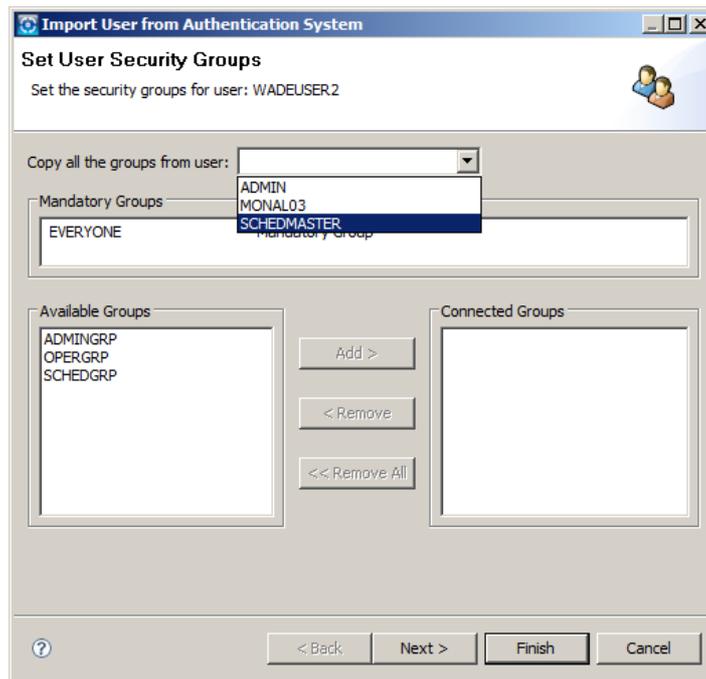


When you next use the Desktop Client's *Security* tab you will be presented with both the locally defined users as well as users defined in the LDAP directory as shown in the example below...



Importing LDAP users

To import an LDAP user, right mouse click on the user and select **Import User from Authentication System**.



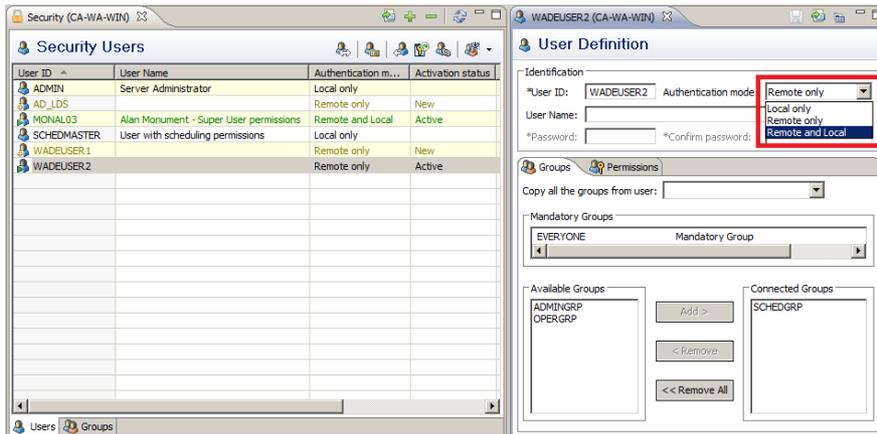
Once imported the user's *Activation Status* column will change from *New* to **Active**. On initial import the default *Authentication mode* for a newly imported LDAP user is *Remote Only*, as shown in the example below.

User ID	User Name	Authentication m...	Activation status	Action required
ADMIN	Server Administrator	Local only		
AD_LDS		Remote only	New	Import user from authentication system
MONAL03	Alan Monument - Super User permissions	Remote and Local	Active	
SCHEDMASTER	User with scheduling permissions	Local only		
WADEUSER 1		Remote only	New	Import user from authentication system
WADEUSER 2		Remote only	Active	

Changing a user's Authentication mode

You can now edit a user's Authentication mode setting by double mouse clicking on the User in the Desktop Client's Security browser and you will be presented with a User Definition pane for that user. The Authentication mode can then be changed using the drop down menu as shown below...

CA Workload Automation DE (dSeries Edition) Cookbook



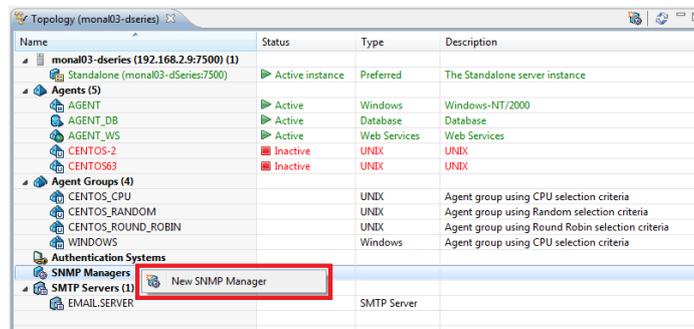
There are 3 *Authentication mode* settings:

- **Local only** – authenticates using the CA WA DE server’s security settings only
- **Remote only** - authenticates via the LDAP server only
- **Remote and Local** - authenticates via the LDAP server first, and if not available, then via the CA WA DE server

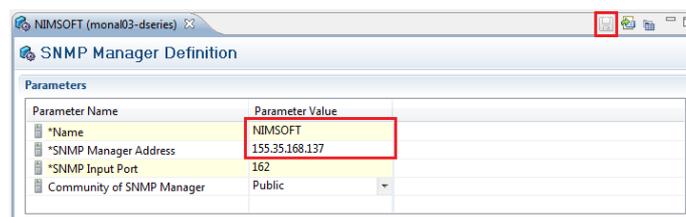
Configuring CA WA DE to work with Nimsoft

Contributed by Alan Monument

Configuring CA WA DE to work with Nimsoft couldn’t be any easier. Simply select the *Admin* perspective button on the Desktop Client and double click the *Topology* item in the left hand pane. The following Topology browser will be displayed...

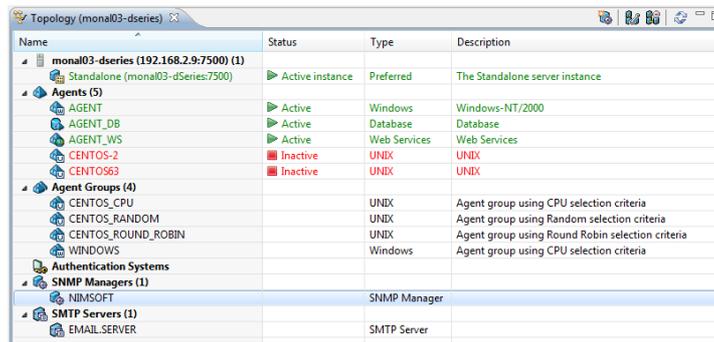


Right mouse click the *SNMP Managers* item in the browser as shown above and select the *New SNMP Manager* menu item. A new window will appear requesting information about the Nimsoft Manager. Simply give the new SNMP Manager a name (such as Nimsoft) and specify the Nimsoft host’s address and then click the *Save* icon...



CA Workload Automation DE (dSeries Edition) Cookbook

A new manager appears in the Topology browser and any SNMP traps being sent by CA WA DE jobs will now be sent to this manager...



Name	Status	Type	Description
mona03-dseries (192.168.2.9:7500) (1)			
Standalone (mona03-dSeries:7500)	Active instance	Preferred	The Standalone server instance
Agents (5)			
AGENT	Active	Windows	Windows-NT/2000
AGENT_DB	Active	Database	Database
AGENT_WS	Active	Web Services	Web Services
CENTOS-2	Inactive	UNIX	UNIX
CENTOS3	Inactive	UNIX	UNIX
Agent Groups (4)			
CENTOS_CPU		UNIX	Agent group using CPU selection criteria
CENTOS_RANDOM		UNIX	Agent group using Random selection criteria
CENTOS_ROUND_ROBIN		UNIX	Agent group using Round Robin selection criteria
WINDOWS		Windows	Agent group using CPU selection criteria
Authentication Systems			
SNMP Managers (1)			
NIMSOF		SNMP Manager	
SMTP Servers (1)			
EMAIL_SERVER		SMTP Server	

The following topics are a collection of articles taken from the **Latest Knowledge Base Updates** section of the CA Workload Automation DE Product Page on support.ca.com (see CA WA DE Product Home Page on page 93) – this section of the support site is recommended reading as it will contain more up to date information than published here.

Workload Automation MIB

When using third-party SNMP managers, such as NAGIOS or similar, you can import a Management Information Base (MIB) description file that defines the format that the SNMP manager uses. The CA Workload Automation DE MIB is located in the Workload Automation default agent directory.

Replacement of server.log Files with stdout.txt Files on UNIX

Each time you restart the CA WA server on UNIX, the server backs up its stdout.txt file. In previous releases, the server accumulated server.log files. To clear stdout.txt files, you must delete them manually from the LogFiles directory where the server is installed.

Note: For more information about clearing stdout.txt files, see the *Admin Perspective Help*.

Validate Database Connection Utility

The CA WA server is now installed with the validatedbconnection utility. You can run the validatedbconnection utility to verify if the database is active. The utility validates the database connection parameters defined in the server configuration.

Note: For more information about the validatedbconnection utility, see the *Implementation Guide*.

Check Integrity of Server Patch Files Utility

The CA WA server is now installed with the MD5Checksum utility. You can run the MD5Checksum utility to check the integrity of the server patch files downloaded from CA websites. You can use the utility as a verification step before applying a patch to the server install.

Note: For more information about the MD5Checksum utility, see the *Implementation Guide*.

Cleanup Utility

The CA WA server is now installed with the cleanup utility. You can run the cleanup utility to delete the server log files that accumulate over time. You can use the cleanup.bat utility for Windows and the cleanup.sh utility

for UNIX.

Note: For more information about the cleanup utility, see the *Admin Perspective Help*.

Changing the SNMP Default Port

If you change the default SNMP port in the agent's `agentparm.txt` file, you must also configure the server shared parameters in the Admin perspective. You must update the Topology by modifying the SNMP Manager settings.

Note: You must restart the server and the agent for the new SNMP port to take effect. For more information about the SNMP Manager settings, see the *Admin Perspective Help*.

Support for Web Services

You can install the web services to let your programmers integrate CA WA in a Service Oriented Architecture (SOA). Your programmers can program any software application that is configured to work with web services to trigger and control CA WA workload. The web services use Simple Object Access Protocol (SOAP) over HTTP or HTTPS.

Note: For more information about installing the CA WA web services, see the *Implementation Guide*.

Limit Size of Spool File in Email Attachments

You can configure a shared parameter to limit the size of spool file attachments in email when providing email notifications for jobs.

Note: For more information about server shared parameters, see the *Admin Perspective Help*.

Configure the Maximum Number of Active Application Generations

By default, the CA WA server now sends email and SNMP notifications when the number of active Application generations reaches 500 (warning threshold). When the number of active Application generations reaches 600, the server shuts down (shutdown threshold).

Note: In previous releases, the server estimated the maximum number of generations that could be processed and sent notifications when the number of active generations reached 75% of this limit and shut down when the number of active generations reached 85% of this limit. In r11.1, the server uses absolute values for the warning and shutdown thresholds.

To configure the warning threshold, set the following property in the `espresso.properties` file:

```
active.appl.gen.notify=
```

To configure the shutdown threshold, set the following property in the `espresso.properties` file:

```
active.appl.gen.stop=
```

Important! If the server shuts down after reaching the shutdown threshold, you must increase the value of `active.appl.gen.stop` before restarting the server. Otherwise, the server will shut down immediately when you try to restart it.

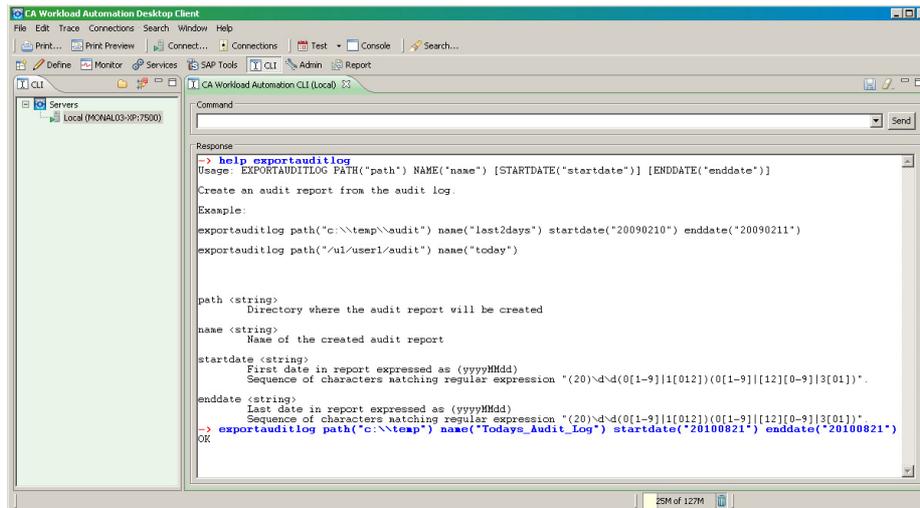
Note: For more information about configuring these properties, see the *Admin Perspective Help*.

Workload Automation Audit Log Scenario

Contributed by Alan Monument

This topic demonstrates via screen shots a number of activities undertaken by two different Desktop Client users and examines the associated entries in the Workload Automation Audit Log that automatically get generated for each function performed.

Workload Automation's Audit Log file can be exported at anytime, for any given date range, by running a Command Line Interface (CLI) command called **exportauditlog**. This command can be run from a scheduled job or from the Command Line Interface in the Desktop Client as shown here...



In the above example, the command:

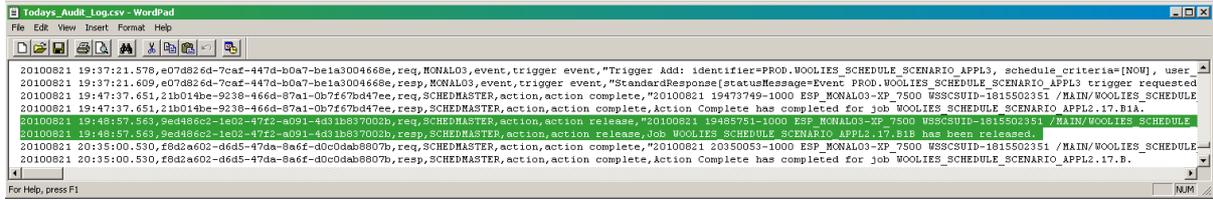
```
exportauditlog path("c:\\temp") name("Todays_Audit_Log") startdate("20130612") enddate("20130612")
```

generates files in both comma separated values (CSV) and HTML formats. For each function performed by Workload Automation there are 2 records generated in the Audit Log; a **request** and a **response** record. Each of these records is divided into seven columns:

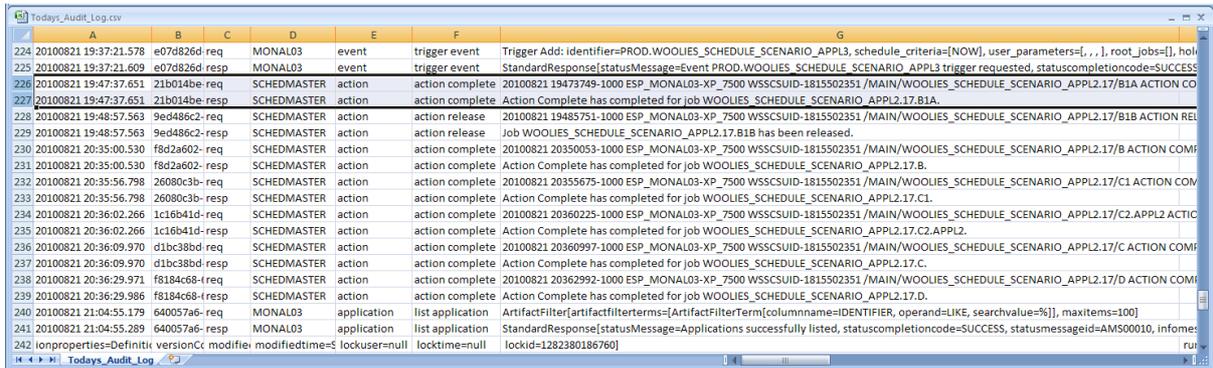
1. A **Date & Time Stamp** (i.e 20100821 19:48:57.563)
2. A **Correlation ID** (this is a system generated value and is the same in both the request and the response records)
3. The **Request / Response** identifier (req or resp)
4. The **User ID** of the person performing the function
5. The **Command Type** (application, action, event,...)
6. The **Command Name** (action release for example to release a job)
7. **Command / Results** (details about the object in question in the request record, and the results of the function in the response record)

The Audit Log file can be viewed as a sequential file...

CA Workload Automation DE (dSeries Edition) Cookbook

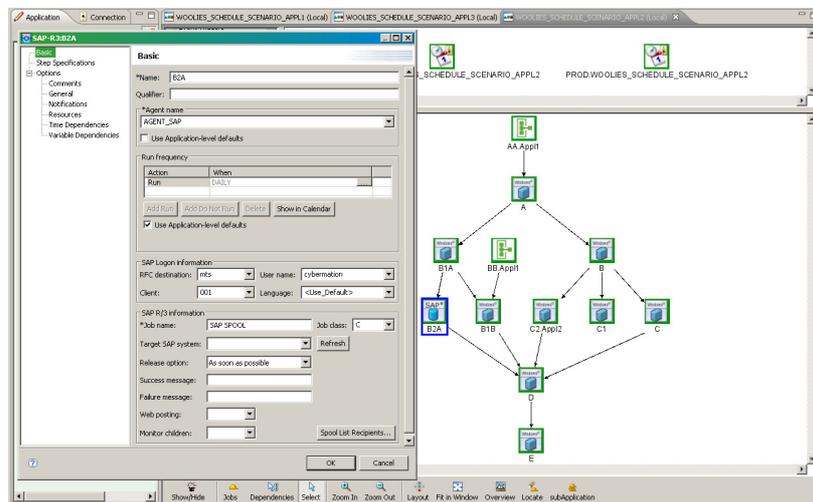


...or opened in an Excel spreadsheet where the CSV data will be organised into columns, which can then be easily sorted or manipulated.



The Audit Log file can also be read by report writing software to produce reports to suit various business requirements. Let's now explore what gets generated in the Audit Log when a number of typical Workload Automation functions are invoked.

Below I have already logged into Workload Automation through the Desktop Client as user MONAL03 and downloaded an application:



Audit Log records are written to record both of these activities...

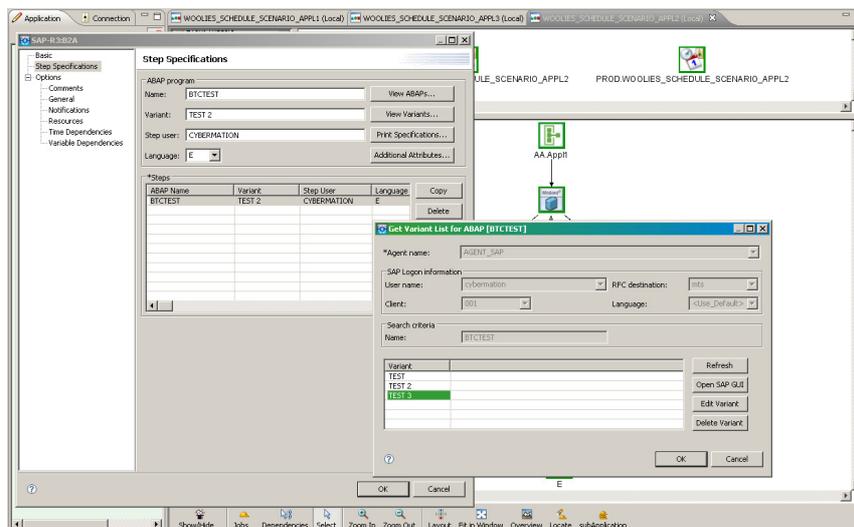
```
20100821 18:37:20.862, ..., req, MONAL03, action, Connect, Connect to Workstation
20100821 18:37:20.862, ..., resp, MONAL03, action, Connect, ESPWSS2836I Logged on successfully to
application server.
:
:
20100821 19:27:27.960, ..., req, MONAL03, application, retrieve application,
WOOLIES_SCHEDULE_SCENARIO_APPL2
```

CA Workload Automation DE (dSeries Edition) Cookbook

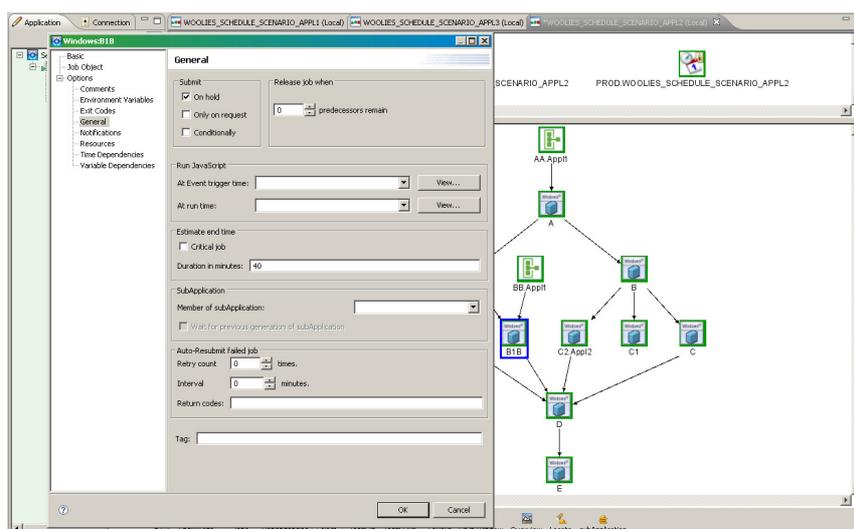
```
20100821 19:27:27.976, ..., resp, MONAL03, application, retrieve application,
"StandardResponse[statusMessage=Application WOOLIES_SCHEDULE_SCENARIO_APPL2 successfully read, ...
[The application's XML definition has been removed for clarity]
```

For the download (or retrieve application) activity logged above, the associated application's XML definition is also recorded in the log but has been removed here for clarity.

After downloading the application to my workstation I double click on the SAP R3 workload object for job B2A (above) to open the SAP R3 workload object dialog window in anticipation of changing the ABAP variant used by this job. To do this I simply click on the *Step Specifications* tab in the left hand pane and the SAP R3 workload object dialog window will now display the Step Specifications editor, which shows that we have currently defined this job to run SAP ABAP **BTCTEST** with a variant of **TEST 2**. By clicking on the *View Variants ...* button the the SAP Agent lists the variants for this BTCTEST ABAP directly from the SAP server – in this example the alternate variant, **TEST 3** will be selected for use..

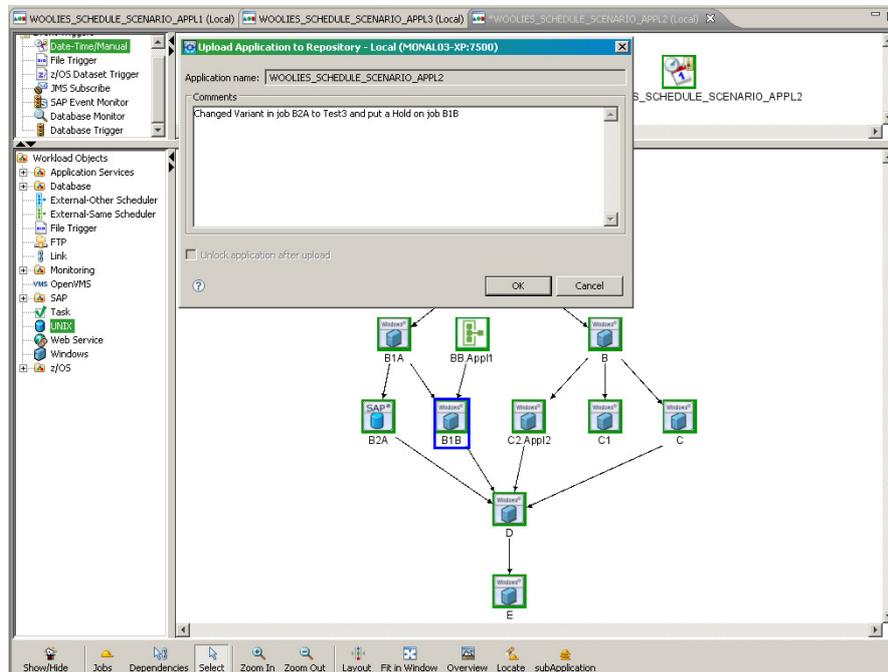


Next a change will be made to another job. The Windows workload object job **B1B** is edited and the *General* tab in the Windows Workload Object dialog window is selected. Selecting the **On Hold** check box will make this job go into a Manual Hold condition when the application is run...



CA Workload Automation DE (dSeries Edition) Cookbook

To implement both these changes, the application is **Upload**-ed to the Workload Automation repository and provided with some commentary regarding the changes made so that these are recorded in the version history associated with this application...



When this application is uploaded, Audit Log records are written together with the entire application definition. Appropriate parts of the definition for the modified jobs B1B and B2A are shown below highlighting the changed data. Note that the Audit Log records show the User ID of the person making the change. Also the comments that were entered during the Upload above are also logged here to complete the documentation of the change.

```
20100821 19:35:50.683, ..., req, MONAL03, application, replace application,
"WOOLIES_SCHEDULE_SCENARIO_APPL2 ...
[Some of this application's XML definition has been removed for clarity]
<app:version>11.1</app:version>
  <app:nt_job name="B1B" >
    <app:hold/>
  </app:nt_job>
  <app:sap_job name="B2A" >
    <app:abapname>BTCTEST</app:abapname>
    <app:variant>TEST 3</app:variant>
    <app:mailaddresslist>
      <app:mailaddress>alan.monument@ca.com</app:mailaddress>
    </app:sap_job>
</app:appl>"
20100821 19:35:50.730, ..., resp, MONAL03, application, replace application,
"StandardResponse[statusMessage=Application WOOLIES_SCHEDULE_SCENARIO_APPL2 successfully
updated, ..., versionComments=Changed Variant in job B2A to Test3 and put a Hold on job B1B,
modifieduser=MONAL03, modifiedtime=Sat Aug 21 19:35:50 EST 2010, ...
```

The selected part of the log above shows the On Hold change to job B1B and the variant change to SAP job B2A. The mailaddress entry in the SAP job B2A is also highlighted as this will result in an e-mail being sent to the specified user when job B2A runs later.

Later in the log you will see that the application was manually triggered by user MONAL03...

```
20100821 19:37:15.437, ..., req, MONAL03, event, trigger event, "Trigger Add:
identifier=PROD.WOOLIES_SCHEDULE_SCENARIO_APPL2, schedule_criteria=[NOW], user_parameters=[, ,
], root_jobs=[], hold=false, triggered_by=null, variables={}
20100821 19:37:15.468, ..., resp, MONAL03, event, trigger event, "StandardResponse[statusMessage=Event
PROD.WOOLIES_SCHEDULE_SCENARIO_APPL2 trigger requested, statuscompletioncode=SUCCESS, ...
```

CA Workload Automation DE (dSeries Edition) Cookbook

When the application runs it's jobs appear in any of Desktop Client's Monitor perspective Custom Views that display jobs that have not yet completed. In the Desktop Client views example below we see running jobs in green, jobs waiting to run in yellow, error conditions in red (not shown), and jobs in exceptional conditions in other colours. Notice that job **B1B** is as requested in the **MANHOLD** (manual hold) state and is highlighted purple.

#	Application Name	Job Name	Job Q.	Anticipated End Time	Average Execution Time	Condi...	State	Status
1	WOOLIES_SCHEDULE_SCENARIO_APP1	AA1	APPL1	Sat Aug 21 21:33:30...	150.0		EXEC	Executing at AGENT
2	WOOLIES_SCHEDULE_SCENARIO_APP1	BB	APPL1	Sat Aug 21 22:33:30...	55.0		PREDWAIT	
3	WOOLIES_SCHEDULE_SCENARIO_APP1	CC ETC		Sat Aug 21 23:43:30...	70.0		PREDWAIT	
4	WOOLIES_SCHEDULE_SCENARIO_APP2	B		Sat Aug 21 21:39:30...	120.0		EXEC	Executing at AGENT
5	WOOLIES_SCHEDULE_SCENARIO_APP2	B1A		Sat Aug 21 19:59:30...	120.0		EXEC	Executing at AGENT
6	WOOLIES_SCHEDULE_SCENARIO_APP2	B1B		Sat Aug 21 21:14:30...	40.0	Predwait	MANHOLD	
7	WOOLIES_SCHEDULE_SCENARIO_APP2	B2A		Sat Aug 21 20:19:30...	20.0		PREDWAIT	
8	WOOLIES_SCHEDULE_SCENARIO_APP2	BB		Sat Aug 21 20:33:30...	55.0		PREDWAIT	
9	WOOLIES_SCHEDULE_SCENARIO_APP2	C	APPL1	Sat Aug 21 23:39:30...	120.0		PREDWAIT	
10	WOOLIES_SCHEDULE_SCENARIO_APP2	C1		Sat Aug 21 22:09:30...	30.0		PREDWAIT	
11	WOOLIES_SCHEDULE_SCENARIO_APP2	C2		Sat Aug 21 23:09:30...	90.0		PREDWAIT	
12	WOOLIES_SCHEDULE_SCENARIO_APP2	D	APPL2	Sun Aug 22 01:39:3...	120.0		PREDWAIT	
13	WOOLIES_SCHEDULE_SCENARIO_APP2	E		Sun Aug 22 01:40:3...	1.0		PREDWAIT	
14	WOOLIES_SCHEDULE_SCENARIO_APP3	AAA		Sat Aug 21 21:56:21...	19.0		PREDWAIT	
15	WOOLIES_SCHEDULE_SCENARIO_APP3	AAA1...		Sun Aug 22 00:26:2...	150.0		PREDWAIT	
16	WOOLIES_SCHEDULE_SCENARIO_APP3	C2	APPL2	Sat Aug 21 21:37:21...	120.0		PREDWAIT	

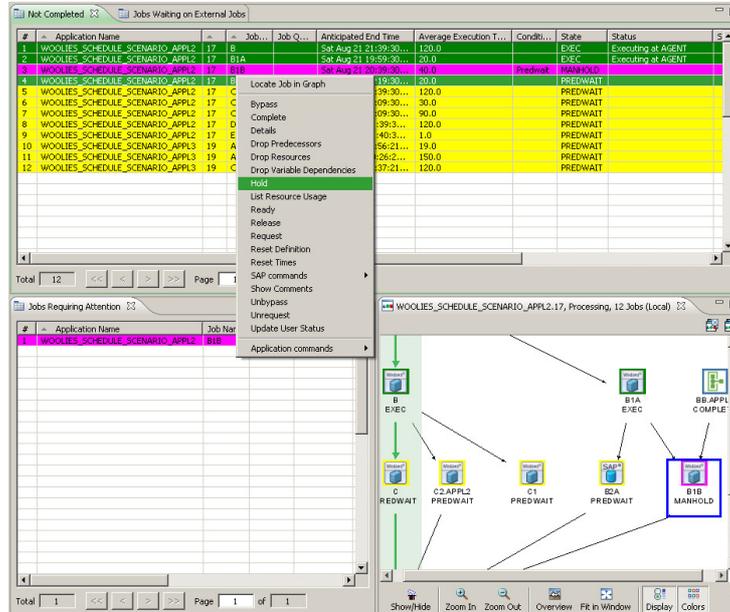
Also note that in the top custom view the *Anticipated End Times* and *Average Execution Times* are conveniently displayed for each job based on that job's previous run time history.

We will let the Application run and satisfy its dependencies on jobs in the application. Notice below that we had previously right mouse clicked on the MANHOLD job and asked to show the job in a graph, which is now displayed in the bottom right pane with the job highlighted in the overall application structure. Predecessor jobs are still running and in this example we shall wait a while before releasing the **B1B** job currently on hold.

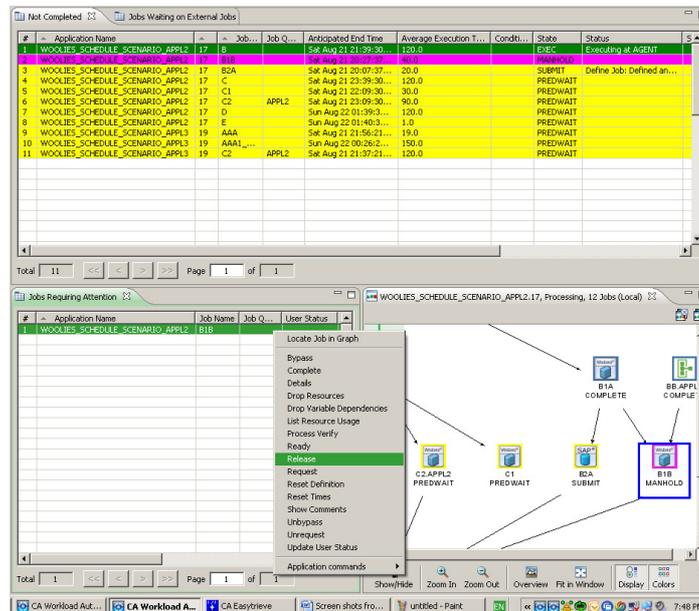
#	Application Name	Job Name	Job Q.	User	Status	State	Anticipated End Time	Average Execution Time	Resource
1	WOOLIES_SCHEDULE_SCENARIO_APP2	B1B				MANHOLD	Sat Aug 21 21:14:30...	40.0	

CA Workload Automation DE (dSeries Edition) Cookbook

Earlier you saw how to define a job *On Hold* before having the application scheduled. This job will always be submitted on hold, unless we again change the job definition. There may also be times when we need to hold any job on an ad hoc basis – this is easy from the Monitor perspective, it is just a case of right mouse clicking on the desired job in the textual or graphical views and selecting the *Hold* option. It is also easy to hold the entire application by right clicking on any of the application’s jobs and selecting the *Application Hold* option – there is no need for you to determine what jobs need to be held, Workload Automation simply puts on hold all other jobs in the application that have not yet started.



Now is the time to release the Held job as all the job’s predecessor jobs have completed and the business has given the all clear to let it run. You simply right mouse click the job object and select *Release*...

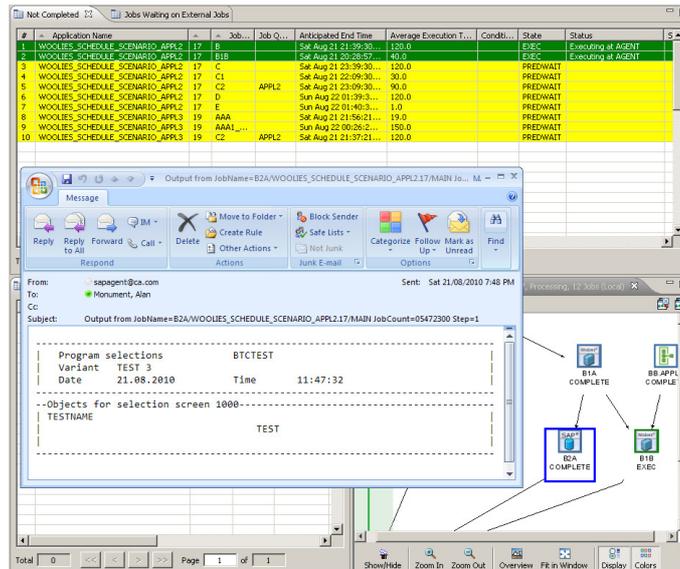


Audit Log records are written when all these functions are performed. Notice that a different User ID is logged for this activity in the log (below) as these jobs are being monitored by user SCHEDMASTER on another workstation. SCHEDMASTER is an operator and has released the job based on instructions from the business.

CA Workload Automation DE (dSeries Edition) Cookbook

```
20100821 19:48:57.563, ..., req, SCHEDMASTER, action, action release, "20100821 19485751-1000  
ESP_MONAL03-XP_7500 WSSCSUID-1815502351 /MAIN/WOOLIES_SCHEDULE_SCENARIO_APPL2.17/B1B  
ACTION RELEASE WSSInfo(10000009,0A8450161D4C) Userid(user.SCHEDMASTER) "  
20100821 19:48:57.563, ..., resp, SCHEDMASTER, action, action release, Job  
WOOLIES_SCHEDULE_SCENARIO_APPL2.17.B1B has been released.
```

Note that Workload Automation functions can only be performed by those users with the appropriate security privileges. SCHEDMASTER may for example be able to perform a *Release* of a PROD application whereas MONAL03 may not. When the SAP R3 job B2A completes, an e-mail gets generated for the people specified in the job definition. Below is an example of the generated email stating that ABAP BTCTEST has now been run using the alternate variant TEST 3 that we selected earlier.



Disaster Recovery (DR) process for the CA WA DE Server

Contributed by Thomas Chou

The following process is one potential way to perform a Disaster Recovery (DR) for CA WA DE. In this scenario we are assuming you have 2x instances of CA WA DE server installed: ABC and XYZ.

- ABC is your **Production** instance; where ABC.com is the host name for the server of that instance
- XYZ is your **Sandbox** instance which will be modified to act as the DR production server in the case of a disaster recovery; where XYZ.com is the host name, and this "Sandbox" server will currently be pointing to a test database

Important: it is assumed that disaster recovery is being taken care of at the database level, and the production server's database is being replicated to the DR location; so that in the event of a disaster at the production data centre you will have an up to date cloned instance of the production CA WA DE database at the DR location.

Steps to switch the Sandbox server to the DR Role

1. Shutdown the XYZ Sandbox instance of the CA WA DE server at the DR data centre.
2. With extreme care modify the following files on XYZ.com in the <CA WA DE installation directory>\conf directory (recommendation: copy and comment out the existing parameters to simplify the rollback later):

- a. **db.properties** - update this file to point to the cloned database - please replace the values in the following parameter examples with appropriate values to access your replicated production database:

- i. DatabaseName: *ESP*
- ii. userid: *WA*
- iii. servename: *XYZ.RDBMS*

- b. **server.properties** - update the GUID of XYZ to match with the GUID defined in server.properties on ABC.com - please replace the values in the following parameter example with appropriate values from your production (ABC.com) instance's server.properties file:

- i. espresso.guid= *192.168.0.9_7500_56B37051-3710-47CB-8D22-AA46EF55945A*

3. Make any required Agent and Agent Group updates to the Cloned Database

- a. If the agents at the Production data centre cannot be reached from the DR data centre, then the following optional step will need be taken to delete existing Agent and Agent Group definitions. You can remove all the Agent and Agent Group definitions from the ESP_AGENT_RP, ESP_AGENTGRP_RP, and ESP_AMS_DF database tables using the following example query:

```
DELETE from ESP_AGENT_RP;  
DELETE from ESP_AMS_DF where ARTIFACTTYPE= 'AGENT' ;  
DELETE from ESP_AGENTGRP_RP;  
DELETE from ESP_AMS_DF where ARTIFACTTYPE= 'AGENTGRP' ;  
COMMIT;
```

Important Note: Agent Groups for Windows & Unix Agents are only available with CA WA DE r11.3 and greater – if you don't use Agent Groups remove the last two DELETE statements.

Note: If you need to perform this step, you will then also need to add new Agent and Agent Group definitions to point to the agents in the DR data centre, after the XYZ server instance has been started.

Important note: Ensure you do regular backups of the Agent and Agent Group definitions on your Production database using the *imexutil* utility so that those Agent and Group definitions can easily be restored back into the production database when the cloned database is eventually rolled back into production after the DR situation has been resolved (see the topic Exporting/Importing Workload Automation Repository Artefacts on page 18 for details).

4. Update the Cloned Database to work with the Sandbox server:

- a. Update the ESP_CONFIG_PROPERTY database table to use the Sandbox server using the following example query:

```
UPDATE ESP_CONFIG_PROPERTY set IN_USE_1= 'XYZ.com' ,  
IN_USE_2=XYZ.com' , DESIRED=XYZ.com' where CONFIG_GROUP_ID='
```

CA Workload Automation DE (dSeries Edition) Cookbook

```
<config-group-id of ABC>' AND NAME='espresso.local.host';  
COMMIT;
```

Replace *XYZ.com* with the host name of the sandbox CA WA DE server. Replace *Config-group-id of ABC* with the value found in the ESP_CONFIG_GROUP table (i.e. 101 as in this example below)...

	ID	NAME	TYPE
1	101	192.168.0.9_7500_56B37051-3710-47CB-8D22-AA46EF55945A	CONFIG_INSTANCE
2	102	#SHARED	CONFIG_SHARED
3	103	EMAIL_SERVER	CONFIG_SMTP
4	200102	APM	CONFIG_SNMP
5	300103	MYSNMP	CONFIG_SNMP

5. Start up the Sandbox CA WA DE instance on XYZ.com. It will now function as a copy of the production environment.
6. Add any required Agent and Agent Group definitions before starting the workloads

Hint: During your first DR exercise you might want to use the *imexutil* utility mentioned earlier to export any Agent and Agent Group definitions that you create for the DR environment, so these can be quickly imported during any future DR situation.

Hint: See also the topic *Skipping non-essential workloads during Disaster days* on page 72.

Returning to normal operations

When returning to normal operations, perform the following:

1. Shutdown the DR (Sandbox) instance of the CA WA DE server at the DR data centre – see also *Steps to switch back to the Sandbox Role* below.
2. Backup the cloned CA WA DE database at the DR data centre.
3. Restore the CA WA DE database into the production data centre.
4. If you previously needed to redefine Agents in the DR data centre instance, you can remove all the Agent definitions from the ESP_AGENT_RP, ESP_AGENTGRP_RP, and ESP_AMS_DF database tables using the following example query:

```
DELETE from ESP_AGENT_RP;  
DELETE from ESP_AMS_DF where ARTIFACTTYPE='AGENT';  
DELETE from ESP_AGENTGRP_RP;  
DELETE from ESP_AMS_DF where ARTIFACTTYPE='AGENTGRP';  
COMMIT;
```

Important Notes: Agent Groups for Windows & Unix Agents are only available with CA WA DE r11.3 and greater – if you don't use Agent Groups remove the last two DELETE statements.

Note that you will also need to restore the original production Agent and Agent Group definitions after the production CA WA DE server has been started, and *before* workloads are restarted.

5. Update the restored Database to work with the Production server

- a. Update the ESP_CONFIG_PROPERTY database table to use the Production server using the following example query:

```
UPDATE ESP_CONFIG_PROPERTY set IN_USE_1='ABC.com',  
IN_USE_2='ABC.com', DESIRED='ABC.com' where CONFIG_GROUP_ID='  
<config-group-id of ABC>' AND NAME='espresso.local.host';  
COMMIT;
```

Replace *ABC.com* with the host name of the Production CA WA DE server. Replace *Config-group-id of ABC* with the value found in the ESP_CONFIG_GROUP table.

6. Start up the Production CA WA DE instance on ABC.com.
7. Use the *imexutil* utility to import a previously exported set of production Agent and Agent Group definitions (see the topic Exporting/Importing Workload Automation Repository Artefacts on page 18 for details), using command similar to:

```
importagent -dir //WAExport/Agents  
importagentgrp -dir //WAExport/AgentGrps
```

Steps to switch back to the Sandbox Role

1. Shutdown the XYZ sandbox instance of the CA WA DE server at the DR data centre.
2. Take extreme care and modify the following files on XYZ.com in the <CA WA DE installation directory>\conf directory:
 - a. **db.properties** – update this file to point back to the original Sandbox database.
 - i. Uncomment/change what you had commented/changed previously
 - b. **server.properties** - update the guid back to point back to the original Sandbox guid.
 - i. Uncomment/change what we had commented/changed previously

Hint: Simply comment those changes you made earlier to make this the DR server so that these settings can be used again in future DR exercises; and un-comment the previous Sandbox settings to return this server to its original Sandbox environment settings.

3. Start up the CA WA DE instance on XYZ.com - it will now function as the Sandbox environment again.

CA WA System Agents and Plug-Ins

Modifying Linux Firewall to cater for WA System Agents

Contributed by Alan Monument

Issue the following commands to modify the Linux firewall to allow CA WA System Agents to communicate with their CA WA DE Management Server running on a Linux platform:

```
# service iptables save
# service iptables stop
# chkconfig iptables off
# system-config-firewall-tui (see below for instructions to add port 7520:tcp to "Other Ports")
# service iptables start
# chkconfig iptables on
```

You add port 7520:tcp to "Other Ports" in the system-config-firewall-tui as follows: press the **space bar** to place an asterisk (*) in the *Firewall: Enabled* checkbox and select **Customise**; reply **Forward** on *Trusted Services*; tab to the **<Add>** item on *Other Ports* and press the **Enter** key; on the *Port and Protocol* page type **7520** into *Port/Ports Range* and **tcp** into *Protocol*, and select **OK**, followed by **Close**; select **OK** on *Firewall Configuration*, and reply **Yes** to the *Warning* dialog.

Preventing Jobs from running under a "root" user id

When a System Agent starts, it checks to see if local security is turned on. If local security is turned on (security.level is set to "on" in the agentparm.txt file), the System Agent then checks to determine whether a security file (named *security.txt*) resides in the System Agent installation directory.

The supplied security.txt file has the following contents:

```
c a * * * (agent commands are allowed from all ESP hosts by all user ids)
f d * * + (FTP commands are disallowed from all ESP hosts by all user ids)
x d * * + (execute commands are disallowed from all ESP hosts by all user ids)
```

If the security file does not exist, default security rules apply which disallows all jobs from running with a root user id. If the security file is non-existent the following rules apply:

```
x a * * + (execute commands are allowed from all ESP hosts by all user ids)
x d * root + (execute commands are disallowed from all ESP hosts by root user id)
c a * * *
f a * * + (FTP commands are allowed from all ESP hosts by all user ids)
```

You should either rename the supplied security file to adopt the above default rules, or modify it as follows to enable all user ids with the exception of root to execute scripts or commands on the agent:

```
c a * * *
f d * * + (FTP commands are disallowed from all ESP hosts by all user ids)
x a * * +
x d * root +
```

CA Workload Automation DE (dSeries Edition) Cookbook

Further details can be found in *Setting up local security on ESP Agent* in the *Setting up security* section of the *System Agent Administrator's Guide*.

Configuring System Agent using the “silent install” configuration file

You can make organisation specific overrides to default parameters in the System Agent “silent install” *installer.properties* configuration file. Below are some common examples. Simply locate the following text in the configuration file and change the highlighted text to suit your organisational requirements.

```
#=====#
#
#           Agent Configuration Parameters           #
#
# AGENT_INFO_1 = Agent name (default is AGENT)      #
# AGENT_INFO_2 = Input port (default is 7520)       #
# AGENT_INFO_3 = Encryption key (default is 1234567890ABCDEF) #
#
#=====#
# AGENT_INFO_1= specifies a unique system agent's name using (up to 16 case-
# sensitive characters)
AGENT_INFO_1=<enter.unique.agent.name.here>
AGENT_INFO_2=7520
AGENT_INFO_3=1234567890ABCDEF
#=====#
#
#           Local Security                           #
#
# To enable local security, set LOCAL_SECURITY = 1  #
# Otherwise, set LOCAL_SECURITY = 0                 #
#
#=====#
LOCAL_SECURITY=1
#=====#
#
#           Manager Configuration Parameters         #
```

CA Workload Automation DE (dSeries Edition) Cookbook

```
# #
# You can configure up to three managers at installation time. #
# #
# To configure two managers, uncomment NUM_MANAGER_2=2 (delete #
# the # symbol beside the parameter) and specify the #
# configuration values for MANAGER_1_INFO and MANAGER_2_INFO. #
# #
# To configure three managers, uncomment NUM_MANAGER_3=3 #
# (delete the # symbol beside the parameter) and specify the #
# configuration values for MANAGER_1_INFO, MANAGER_2_INFO, and #
# MANAGER_3_INFO. #
# #
# MANAGER_1_INFO_1 = Manager ID (default is CENTRAL_MANAGER) #
# MANAGER_1_INFO_2 = Manager IP address #
# MANAGER_1_INFO_3 = Manager port (default is 7507) #
# #
# MANAGER_2_INFO_1 = Manager ID (default is CENTRAL_MANAGER2) #
# MANAGER_2_INFO_2 = Manager IP address #
# MANAGER_2_INFO_3 = Manager port (default is 8507) #
# #
# MANAGER_3_INFO_1 = Manager ID (default is CENTRAL_MANAGER3) #
# MANAGER_3_INFO_2 = Manager IP address #
# MANAGER_3_INFO_3 = Manager port (default is 9507) #
# #
#=====#
NUM_MANAGER_2=2
#NUM_MANAGER_3=3
# This is the Production Manager
MANAGER_1_INFO_1=ESP_PROD_7500
MANAGER_1_INFO_2=esp.prd.com.au
MANAGER_1_INFO_3=7507
```

This is the Development Manager

MANAGER_2_INFO_1=ESP_DEV_7500

MANAGER_2_INFO_2=esp.dev.com.au

MANAGER_2_INFO_3=7507

MANAGER_3_INFO_1=

MANAGER_3_INFO_2=

MANAGER_3_INFO_3=

#####

#

Agent System Log Files

#

log.folder= is the location of the log files. The default

folder is log, which resides in the directory

that contains the system agent files.

log.levels 0 (default), 1, and 2 provide logs of any errors

including the receiver and transmitter logs.

log.maxsize= is the maximum log size in bytes.

log.archive=0 (default) appends current date and time to the

log file

log.allow.method=true (default) Logs additional debugging

messages in all log files.

#

#=====

Log.folder=/log/esp/spool

log.level=0

log.maxsize=1024000

log.archive=0

log.allow.method=true

#####

#

Log Expiration parameter

```
# #
# runnerplugin.spool.clean.enable=true enables the spool file #
# cleaner which deletes spool files based on #
# following rules (default = false) #
# runnerplugin.spool.expire=40D agent deletes spool files that #
# are older than 40 days (default is 10 days) #
# runnerplugin.spool.sleep=1D every 1 day (the default) the #
# agent checks for spool files that meet the #
# expiration time and deletes them. #
# #
#=====#
runnerplugin.spool.clean.enable=true
runnerplugin.spool.expire=40D
runnerplugin.spool.sleep=1D
#####
# #
# Operating System dependant parameters #
# #
# oscomponent.loginshell=true indicates how to invoke the Shell #
# program when executing a script. Invokes the #
# shell as a login shell if you specify true. #
# Default is false. Note: If you specified true, #
# the shell program looks in the directory #
# specified by the HOME environment variable, and #
# tries to execute the user's login scripts (in #
# addition to the .cshrc script). #
# #
#=====#
oscomponent.loginshell=true
```

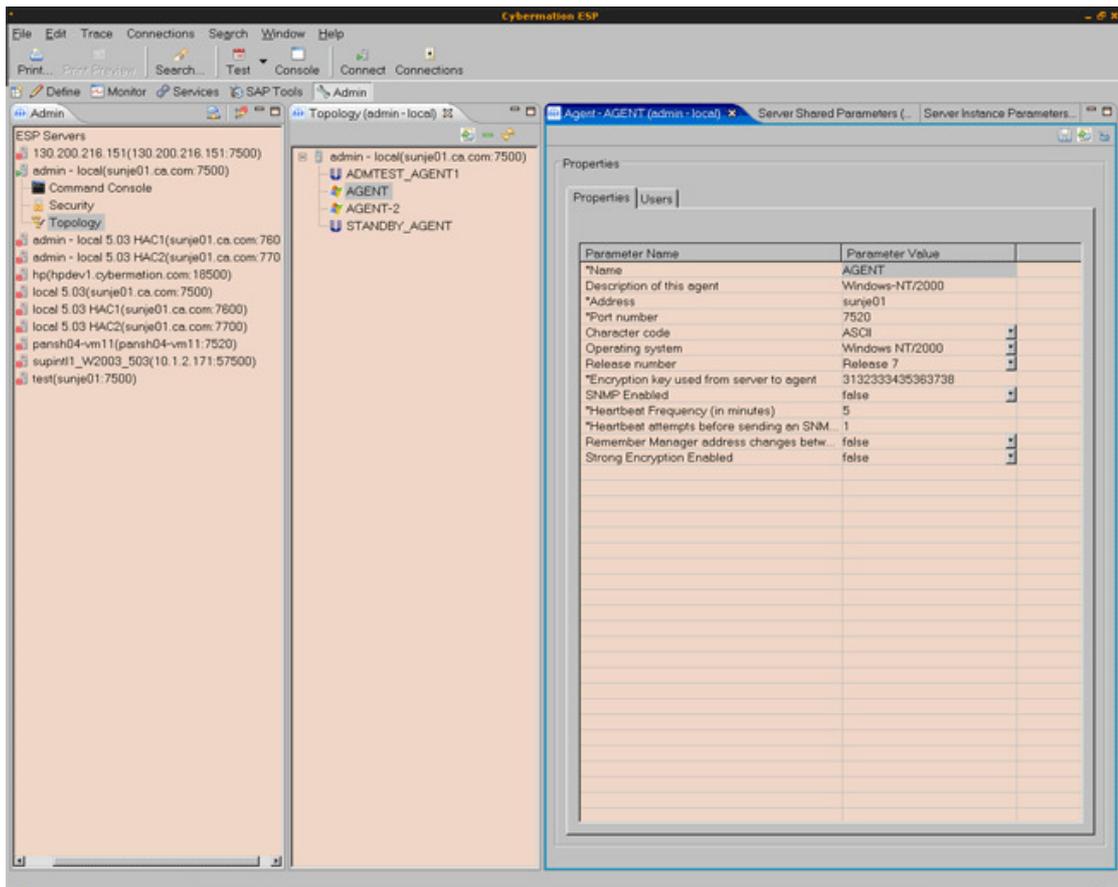
Quick self-check for DE Server and Agent connectivity issues

CA Tech Note: Document ID: TEC436329

It's recommended you go through the following checklists to quickly pinpoint most of the causes of server and agent connectivity issues:

Checks on agent side:

- Get the following topology definition for the agent in desktop client GUI.



- Get the file agentparm.txt from agent installation directory.
- Things to check:
 - Make sure the agent has been started by checking if the process id shows in the file status.os under agent installation directory
 - If a job gets stuck in a READY state in the Desktop Client, the agent has not been able to understand and respond to the CA WA DE manager's request; so make sure the values for the following parameters are identical both in the Desktop Client' topology and in the System Agent's agentparm.txt file:

Topology agent definition	The agent's agentparm.txt file
Name	agentname=
Release number	installer.version=

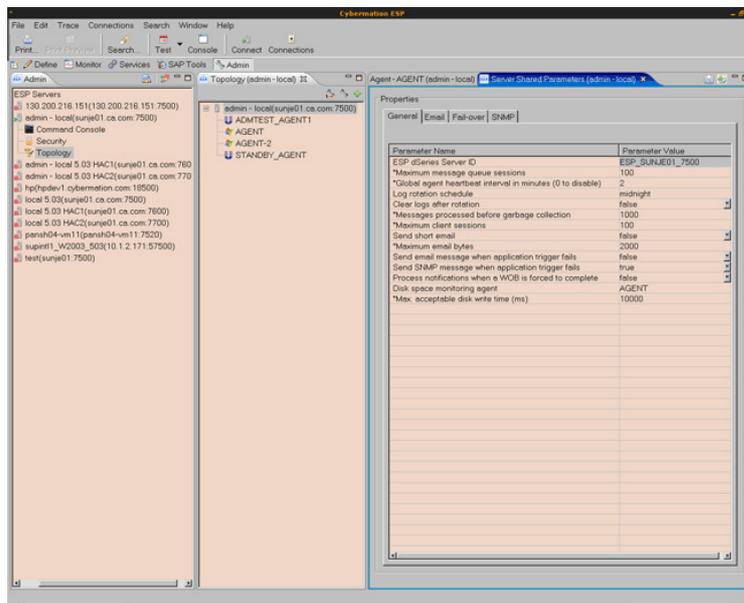
CA Workload Automation DE (dSeries Edition) Cookbook

Encryption key used ...	security.cryptkey= (leave off the 0x)
Port number	communication.inputport=

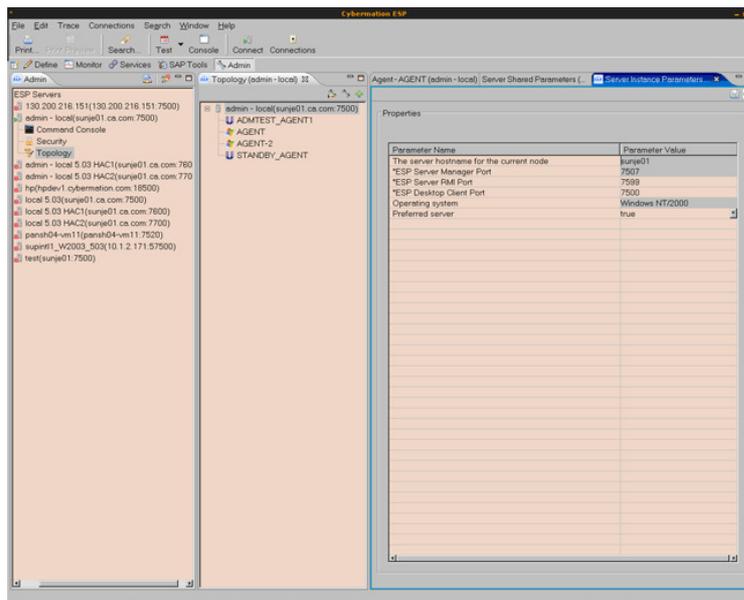
- Make sure the value of parameters "Address" is the hostname or ip address of the agent host.
- Test connectivity from server to agent:
 - On the CA WA DE server host, open a terminal, type the command:
 - telnet <Address> <Port number>
 - Replace the <Address> and <Port number> by the values from topology definition above. If the response is "Connected to...." the connection should be ok.

Check server parameters

- Get the following topology definitions for the CA WA DE server:
 - Server Shared Parameters



- Server Instance Parameters



CA Workload Automation DE (dSeries Edition) Cookbook

- Get the file agentparm.txt from agent installation directory.
- Things to check:
 - Make sure the values for the following parameters are the same:
 - <Topology definition> <agentparm.txt>
 - ESP dSeries Server ID communication.managerid_ =
 - The Server Host Name.... communication.manageraddress_ =
 - ESP Server Manager Port communication.managerport_ =
 - Test connectivity from agent to server:
 - On agent host, open a terminal, type the command:
 - telnet <The Server Host Name> <ESP Server Manager Port>
 - Replace the <The Server Host Name> and <ESP Server Manager Port> with the values from definition topology. If the response is "Connected to...." the connection should be ok.

Using CA WADE with OpenBSD/FreeBSD

Contributed by Alan Monument

The OpenBSD/FreeBSD platform is not directly support by a System Agent and not officially supported by the Remote Execution plug-in; however the following instructions will allow you to use SSH via the Remote Execution plug-in to run commands on this platform. Note that on SSH is implemented by default on this platform.

1. Install a Remote Execution plug-in on a server that can reach the target OpenBSD/FreeBSD server.
2. Under the host System Agent's installation directory, navigate down to the /config/proxy directory. In here you will find a number of <OS>.properties files.
3. In the proxy directory create a new file called OpenBSD.properties with the following contents:

```
host=<hostname>
port=22
protocol=ssh2
type=unix
spoolHome=/home/<user-ID>/spool

default.user=<user-ID>
default.password=9C11675F9ED8322327F5
```

Replace <hostname> with the hostname of the target OpenBSD/FreeBSD server and <user-ID> with a user id on that server with the appropriate permissions to run the work that you want to execute. The encrypted default.password string is created by running the user's password through the CA WADE *password.sh* utility described in the *Encrypting and Changing Passwords* topic of the System Agent's Implementation Guide, and copy & pasting the output into the above properties file.

4. On the target OpenBSD/FreeBSD server log on as the above user id and create the directory specified in the above *spoolHome* parameter. Later you will find the output from jobs run via the Remote Execution plug-in using the above profile, in directories under this directory.

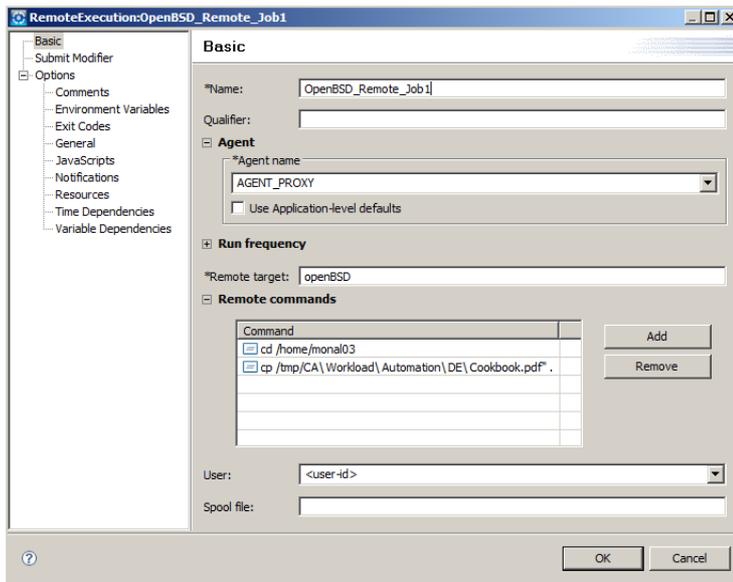
- Using the Desktop Client, use a Remote Execution workload object to define the work you want to run on your OpenBSD/FreeBSD server, where:

Agent name is the name assigned to the Remote Execution plug in you installed.

Remote target is the name of the <OS>.properties file to be used (i.e. openBSD.properties).

Remote commands are the command line commands you want to run in the target OpenBSD/FreeBSD server's SSH session.

User is the <user-ID> defined above.



Hint: Each work Remote Execution workload object can run one or more commands. Consider breaking all the commands you need to run, into more granular logical units of work that you want to manage. The more workload object you use the easier it will be for you to keep track of progress through the job stream.

CA WA DE Desktop Client

The following topics are a collection of articles taken from the **Latest Knowledge Base Updates** section of the CA Workload Automation DE Product Page on support.ca.com (see CA WA DE Product Home Page on page 93) – this section of the support site is recommended reading as it will contain more up to date information than published here.

Disable Browser Debugging for Online Help

If debugging is enabled in your browser, a JavaScript error occurs when you access a help topic in CA WA Desktop Client. You can ignore the errors. To avoid getting JavaScript errors, disable debugging in your browser.

CA WA Desktop Client Workspace

All of your local Applications and other scheduling-related definitions are preserved in the following directory to avoid conflicts with other Eclipse-based products:

```
drive:\Documents and Settings\userid\workspace-CAWA-11.1
```

Security Permissions

We have changed the permissions required for simulation so that a user does not require the EVENTX permission. As long as users have read access to the Event and the corresponding Application, they can simulate the Event.

We have also changed the permissions required to display SAP job logs and spool files so that a user does not require access to all commands. You can now grant users access to the commands to display logs and spool files and deny them access to the other commands with the following APPLX permissions:

```
APPLX.*.*.GETSPOOLFILE  
APPLX.*.*.SUBSCRIBE
```

We have added a permission named VARIABLE that determines a user or group's access to global variables. Global variables let you store information that you can reuse across Applications.

Note: For more information about security permissions, see the *Admin Perspective Help*.

Getting Help when using the Desktop Client

Pressing the F1 key when using the Desktop Client will present you with a context sensitive Help pane – use this to discover more about the fields and options available to you on the desktop. Usually links to associated Help topics are available in the help content. Help system Contents and a Search capability are also available in this Help pane.

Planning Your Site's Naming Conventions

Contributed by Bob Pyette

The naming of the CA Workload Automation objects will vary from company to company. Your organisation may already be using less sophisticated scheduling solutions and already have a naming convention in place that the users and operators are familiar with and it may be smart to continue with that convention or take the opportunity to implement a new one.

CA Workload Automation DE (dSeries Edition) Cookbook

The naming conventions for different entities within Workload Automation ties directly into the associated security permissions identified in the Workload Automation server's access control lists. Regardless of the names chosen for Applications and jobs for monitoring purposes; a user can easily set up custom views which filters information based on any job-related criteria they want.

The following will discuss recommendations for CA Workload Automation naming standards and will provide some examples of naming conventions that other companies have used.

Server Names

Server names can contain any alphanumeric characters and the following special characters: commercial at (@), dollar sign (\$), and underscore (_). However, because CA Workload Automation uses Server names as file names, you must adhere to standard file-naming conventions for your operating system. A name containing a space or dollar sign (\$) for example, is not allowed on UNIX.

Examples of customer implementations have the server named ESP_PROD for a production instance, and ESP_TEST for a test instance. Other customers have used the machines hostname as the server name; however in a High Availability configuration, using the machines host name would not be suitable. Having a name such as ESP_PROD or simply ESP is suggested.

Agent Names

Agent names can contain any alphanumeric characters and the following special characters: commercial at (@), dollar sign (\$), and underscore (_). The Agent name is limited to 16 characters and is required when you configure the CA Workload Automation Server for this Agent. When installing the CA Workload Automation server, the associated installed agent name defaults to Agent.

It is a recommendation and best practice to name the agent on other servers to be the same as the actual machine's hostname.

In the event you want to install multiple agents installed on the same machine, you can qualify the hostname with a unique identifier (i.e. KFXPROD1_1, KFXPROD1_2 etc ...).

Application Names

Application names can contain any alphanumeric characters and the following special characters: commercial at (@), dollar sign (\$), and underscore (_). A meaningful name should be selected for the Application name which can be up to 128 characters long. The same application definition may be used for both production and development by using a variable that resolves to either PRD or DEV in the runtime name of the Application.

Job Names

A meaningful name (up to 128 characters long) should also be selected for Job names. Job names must be unique within an Application. If a job needs to be defined more than once it must be qualified. You can use a qualifier to help make a job name more meaningful.

Event Naming

An Event name has two parts: a prefix and a descriptive name. The prefix allows you to group Events together. You can list Events based on their prefix. For example, all of your production Events could have a prefix of PROD (or PRODUCTION), and all of your test Events could have a prefix of TEST.

Event names can contain any alphanumeric characters and the following special characters: commercial at (@), dollar sign (\$), and underscore (_). The Event name must be unique. The name you specify, combined with the Event prefix, uniquely identifies the Event.

CA Workload Automation DE (dSeries Edition) Cookbook

Normally, this name will be the same as your Application name.

Event Prefix names can contain any alphanumeric characters and the following special characters: commercial at (@), dollar sign (\$), and underscore (_).

Typical implementations of Event Prefixing includes setting up conventions agreed upon that convey certain information. For example, you can create prefixes that represents what environment to run in.

For example:

- PRD.applicationname
- DEV.applicationname

Both of the above events run the same application but the first one is for the production environment and the second one is development.

Event naming is therefore very important when you are planning on implementing applications that are dynamically built based on what environment is triggering it; using a single definition of the application and an Event for each environment.

Calendar Names

Calendar names can contain any alphanumeric characters and the following special characters: commercial at (@), dollar sign (\$), and underscore (_). Each Calendar name must be unique and cannot be more than 30 characters.

Select Calendar names that are meaningful to your users.

Hint: Most customers use only 1 calendar, the SYSTEM calendar, which is predefined.

WORKDAYS

WORKDAY is understood by Workload Automation to represent any of the five standard working week days of Monday through Friday. DAILY is understood to represent all seven days of the week.

Holidays

Specify a meaningful name for the holiday. Holiday names must start with a letter and can contain letters, numbers, and underscores. There is no limit to the number of characters in the name. Characters are not case-sensitive.

Holidays stop jobs from running on a WORKDAY. For example typical Australian holidays would include:

NEW_YEARS_DAY

AUSTRALIA_DAY

GOOD_FRIDAY

EASTER_MONDAY

ANZAC_DAY

QUEENS_BIRTHDAY

BANK_HOLIDAY

LABOUR_DAY

CHRISTMAS_DAY

BOXING_DAY

Special Days

Specify a meaningful name for the Special day. The special day must start with a letter and can contain letters, numbers, and an underscore.

A special day is a user defined day with special significance for scheduling in your particular installation of CA Workload Automation.

Workload Automation has a built-in understanding of many common scheduling terms. Special days are only required to augment the dictionary of terms that Workload Automation understands.

Alerts

Alert names can contain any alphanumeric characters and the following special characters: commercial at (@), dollar sign (\$), and underscore (_). Each Alert name must be unique. An Alert name can be up to 128 characters long.

Alert names should be meaningful and describe the functionality of the alert.

An example of naming convention for Alerts can be to create the Alert name with the application calling it in caps, then the functionality in mixed case followed by ALERT in caps to distinguish that it is an Alert. (i.e. DAILY_RUN_Resubmit_Failed_Job_ALERT)

For example, if you have a generic Alert that force completes the job that issued the alert, you can call it GENERAL_COMPLETE_JOB.

Application-specific Alerts must start with the application's prefix (i.e. CLY). Note that creation and maintenance of these alerts can only be accomplished by those users who are assigned Appropriate security privileges. All other users would have global read access to defined alerts.

Global Alert names must start with GLOBAL. These Alerts can be used by any Application.

Resource Names

Resource names can contain any alphanumeric characters and the following special characters: commercial at (@), dollar sign (\$), and underscore (_). Each Resource name must be unique. A resource name can be up to 32 characters long.

Resource names should be meaningful and describe the resource.

For example if you want to create a resource to represent the availability of an agent, you can create a resource that is simply the agent name. Make sure to use the comments to document what the use of the resource is for.

Application-specific resources must start with the application's prefix (e.g. DAILY_DB_BACKUP). Note that creation and maintenance of these resources can only be accomplished by those users who are assigned to appropriate security privileges. All other users would have global read access to the defined resources.

Global resource names must start with GLOBAL. These resources can be used by any job in any Application.

JavaScript Names

The JavaScript central repository can be used to store JavaScript scripts used in different Applications. This facilitates maintenance when a script used in different Applications needs to be updated. The names of the scripts must be unique and can be up to 64 characters in length. Alternatively, you can create local JavaScript scripts within an Application or Alert and use any name up to 64 characters.

Most customers store all JavaScript scripts in the central repository regardless of how many times it is used. You will then have just one location for ALL scripts being used in your CA Workload Automation environment.

One typical implementation involves defining all JavaScript names in the repository to a certain convention where the Application that will use the JavaScript is in capital letters, and the actual JavaScript action is in mixed case.

For example:

- APPX_Set_Environment is the name of a JavaScript script used by the APPX application to set its variables
- DBA_TURN_ARCHLOG_RESOURCE_OFF is the name of a JavaScript script used in a DBA application to turn a resource off programmatically

Application-specific JavaScript scripts stored in the central repository must start with the application's prefix. Note that creation and maintenance of these JavaScripts can only be accomplished by those users who are assigned with appropriate security privileges. All other users would have global read access to defined JavaScripts.

Role-Based Security

Security

This section provides some general rules with regards to *users* and *groups*.

Each *user* must be defined with a unique user name/userid.

You use *groups* to define the same set of permissions for a group of users. Once you assign permissions to a *group*, you can then associate users with that group. All users in a particular group share the permissions that belong to that group. Groups are useful for users who share common duties and activities.

A *user* can be associated with one or more *groups*. In the case where a user belongs to different groups with different permissions to the same entity, then the user will have the highest level access to that entity. For example, if a user belongs to GROUP1 and GROUP2, and GROUP1 has Alter access to an entity and GROUP2 has Read access to the same entity, then the user will have Alter access to that entity.

User permissions can be used to override the group permissions. For example, if the user belongs to a group that has access to all agents, then access to a subset of those agents can be *denied* through explicit *user permissions*.

User and Group Clean-up

There are two predefined *users* that get installed with the Workload Automation server; ADMIN and SCHEDMASTER.

CA Workload Automation DE (dSeries Edition) Cookbook

IMPORTANT: These users should have their passwords changed to protect against unauthorized usage, as they are detailed in the product documentation.

The following *groups* are also predefined:

- ADMINGRP was designed for those that need to administer the system. This includes adding agents to the topology, configuring the system, issuing system-type commands, setting up security, defining logical resources, and running the VERIFY Application.
- SCHEDGRP was designed to facilitate centralized definition of jobs and related processes.

EVERYONE Group

As new users are defined, they automatically belong to the predefined EVERYONE group. This is a mandatory group and the name of this group cannot be changed.

By default, this group has the following permissions:

- Read access to the SYSTEM calendar, which allows users to use terms defined on the SYSTEM calendar.
- Read access to ADMIN.* for some internal security checking.
- Read access to VARIABLE.DEFAULT.* which allows users to use global variables in the DEFAULT context.

One of the goals is to allow users the ability to display information across the organisation. The general philosophy is to open up what users can actually display within the scheduling environment and lock down what users can actually change in the environment. This will help with further application development and foster re-use of different entities. Users can choose not to list certain entities by using filters in conjunction with the established naming standards.

For example the EVERYONE group could be modified to have the following permissions:

ACL (Access Control List)	Permission	Comments
ADMIN.*	Read	Allows user to use the system
AGENT.*	Read	Access to run jobs on all agents
ALERT.*	Read	Alert definitions for special processing when a job reaches a certain state in processing
APPL.*	Read	Application/job definitions
APPLX.*.*.GETSPOOLFILE	Allow	Allows users to retrieve pool files
APPLX.*.*.SUBSCRIBE	Allow	Allows users to display active Applications/jobs
CALENDAR.*	Read	Calendar definitions
EVENT.*	Read	Allows user to browse and simulate Events
FORECAST.*	Read	Forecast definitions. Note that currently the EVENTX permission is used to control what information can actually be forecast since it relies on the permission for simulation.
JAVASCRIPT.*	Read	JavaScript definitions in central repository

CA Workload Automation DE (dSeries Edition) Cookbook

ACL (Access Control List)	Permission	Comments
RESOURCE.*	Read	Logical resource definitions used within Applications
VARIABLE.DEFAULT.*	Read	Read access to global variables in the DEFAULT context

Group Architecture

In general, you will define groups and associate permissions with groups to facilitate maintenance. Note that each group can have a name up to 20 characters in length and a description up to 40 characters in length.

SYSADMINS Group

You could create a 'super group' called SYSADMINS that includes the privileges for the predefined SCHEDGRP and ADMINGRP groups. This would allow users of this group to have complete access to all CA Workload Automation DE artefacts. The SYSADMINS group would have the following permissions:

ACL (Access Control List)	Permission	Comments
ADMIN.*	Alter	Network topology and security settings
AGENT.*	Allow	Access to run jobs on all agents
AGENTMSG.*	Allow	Issue control commands to agent from Monitor perspective and Command-line console
AGENTUSER.*	Allow	Allow access to run jobs on all agents with any predefined userids
ALERT.*	Alter	Alert definitions for special processing when a job reaches a certain state in processing
APPL.*	Alter	Application/job definitions
APPLX.*	Allow	Control active jobs and Applications
CALENDAR.*	Alter	Calendar definitions
CMD.*	Allow	Command-line console commands
EVENT.*	Alter	Event definitions
EVENTX.*	Allow	Control Events
FORECAST.*	Alter	Forecast definitions. Note that the EVENTX permission is used to control what information can actually be forecast.
JAVASCRIPT.*	Alter	JavaScript definitions in central repository
RESOURCE.*	Alter	Logical resource definitions used within Applications
VARIABLE.*.*	Alter	Global variables in any context

TRAINING Group

It would be worth crating a group called TRAINING that allows users to walk through the Getting Started Guide. The TRAINING group would have the following permissions:

ACL (Access Control List)	Permission	Comments
APPL.QUICK*	Alter	Application/job definitions
APPLX.QUICK.*.*	Allow	Control active jobs and Applications
EVENT.*.QUICK*	Alter	Event definitions
EVENTX.*.QUICK*	Allow	Control Events

Defining Workloads

The following topics are a collection of articles taken from the **Latest Knowledge Base Updates** section of the CA Workload Automation DE Product Page on support.ca.com (see CA WA DE Product Home Page on page 93) – this section of the support site is recommended reading as it will contain more up to date information than published here.

Single Backslash and Double Quotes not recognized

If you specify a text string in a Text File Reading and Monitoring job, or a success pattern in a JMX job, CA WA Desktop Client does not recognize a single backslash or single double quotes. For example, it reads the path `c:\windows\system32\cmd.exe` as `c:windowssystem32cmd.exe`. In this example, you must use an extra backslash to escape each backslash, for example, `c:\\windows\\system32\\cmd.exe`.

Edit Job Dependencies Feature

With Applications that contain a large number of jobs, it becomes difficult to draw, remove, or view dependencies between jobs that are located a large distance from each other in the workspace. The Edit Job Dependencies feature lets you add, remove, or view a job's predecessor and successor dependencies using a dialog instead.

Note: For more information about the Edit Job Dependencies feature, see the *Define Perspective Help*.

Scheduling Based on Time Zones

When you schedule an Event, you can now select <local computer> in the Time Zone drop-down list of the Schedule event dialog, which lets you schedule Events based on the local time zone of the CA WA Desktop Client computer. By default, Event scheduling criteria uses <server>, which is the time zone of your CA WA server.

Note: The specified time zone is only used to evaluate the schedule criteria before it is converted to the server's time zone. The schedule follows the daylight saving changes of the server's time zone, *not* the specified time zone. If the specified time zone and the server's time zone have a different daylight saving period, the scheduled times displayed in CA WA Desktop Client (converted to the local computer's time zone) will vary according to the rules of the server's time zone. For more information about scheduling based on time zones, see the *Define Perspective Help*.

Scheduling an Event on a Particular Day

When scheduling an Event on a particular day, you no longer have to specify the ONCE scheduling term. For example, to schedule an Event only on September 1, 2008, you can specify the following:

Schedule SEPTEMBER 1, 2008

Note: For more information on scheduling Events, see the *Define Perspective Help*.

Scheduling Terms

You can use the following pre-defined scheduling terms in CA WA DE workload definitions:

ALL

Specifies all of the elements within a period (equivalent to ANY and EVERY). For example, you can use FIRST WORKDAY OF ALL MONTHS instead of FIRST WORKDAY OF MONTH.

AM

Specifies the time between 00:00 and up to, but not including, 12:00.

AND

Indicates an optional connecting word that you can use to specify multiple similar scheduling terms (days of the month, months of the year) and ordinal numbers, for example:

FIRST AND LAST WORKDAY OF EACH MONTH

TUESDAY AND WEDNESDAY

JUNE, JULY AND AUGUST

Note: You cannot use this term with times of day.

ANY

Specifies all of the elements within a period (equivalent to ALL and EVERY), for example, ANY WORKDAY OF 6th MONTH OF YEAR.

ANYDAY

Indicates that no day of the week restriction exists. For example, ANYDAY OF JUN specifies all the days in June.

COMPLETE

Indicates that you want the schedule to occur only during a complete period. For example, you can specify FIRST COMPLETE WEEK OF YEAR. If a week starts in one year but finishes in the next year, CA WA interprets the expression to mean the next complete week.

DAILY

Specifies every one day.

DAYS

Specifies a period of time in days, for example, TODAY PLUS 5 DAYS.

EACH

Specifies every occurrence of a particular day or period, for example, EACH MONDAY OF OCTOBER.

ENDING

Specifies the ending point for a repeating schedule, for example, DAILY AT 9AM STARTING TOMORROW ENDING 1JUL2005. This term is identical to the term UNTIL.

Note: You cannot use this term in Run and Do not Run statements for jobs and Applications.

EVERY

Specifies all elements within a period (equivalent to ANY and EACH), for example, EVERY 1 HOUR.

Note: You cannot use this term as a run criteria for a job.

EVERY *n units*

CA Workload Automation DE (dSeries Edition) Cookbook

Indicates how often, in units of time, you want the schedule to recur. The units can be seconds, minutes, hours, workdays, days, weeks, months, or years, for example, EVERY 5 MINUTES, EVERY 1 HOUR, EVERY 2 DAYS.

Note: You cannot use this term as a run criteria for a job.

EXCEPT

Indicates an exception. You can use different criteria as exceptions, for example:

DAILY EXCEPT MONDAY TUESDAY

FRIDAYS EXCEPT FIRST AND LAST DAY OF MONTH

WEEKDAYS EXCEPT WEDNESDAYS

HOLIDAYS EXCEPT CHRISTMAS

WORKDAYS EXCEPT INVENTORY_DAY

Note: You cannot use this term in a run criteria for a job with a date, for example, Run DAILY EXCEPT OCT 3, 2008. However, you can accomplish this using a Do not run statement: Run DAILY; Do not run OCT 3, 2008.

FIRST

Specifies the first occurrence of a particular day or period, for example, FIRST MONDAY OF MONTH or FIRST DAY OF 2nd FISCAL_MONTH OF FISCAL_YEAR. You can also use the ordinal number 1ST, for example, 1ST HOLIDAY OF YEAR.

HOLIDAYS

Specifies all of the holidays in each calendar. You can refer to specific holidays by name, for example, CHRISTMAS.

HOURLY

Specifies every one hour, for example:

HOURLY means 00:00, 01:00, and so on.

HOURLY STARTING AT 2:15 means 2:15, 3:15, and so on.

LAST

Specifies the final occurrence of a particular day or period, for example:

LAST HOLIDAY OF YEAR

LAST WORKDAY OF FISCAL_MONTH

FIRST AND LAST TUESDAY OF MONTH

LESS *n units*

Requests an adjustment by subtracting a value which must be a whole number. For example, you can specify LESS 2 WEEKDAYS, LESS 1 MONTH, or LESS 0 WORKDAYS. To obtain the second last day of the month, you can use LAST DAY OF MONTH LESS 1 DAY.

MIDDAY

Specifies the time 12:00 or noon.

MIDNIGHT

Specifies the time 24:00. CA WA recognizes 24:00 as the same as 00:00. For example, if you specify MIDNIGHT MONDAY, it is the same as 00:00 TUESDAY.

MINUTE/MINUTES

Specifies a period of time in minutes, for example, EVERY 2 MINUTES.

MONTH/MONTHS

CA Workload Automation DE (dSeries Edition) Cookbook

Specifies a period of time in months, for example 1ST MONTH OF YEAR.

MONTHLY

Specifies every one month.

Note: You cannot use this qualifier by itself within a job definition. You can use RUN 15TH MONTHLY, but you cannot use RUN MONTHLY.

NOON

Specifies the time 12:00.

NOW

Specifies the current virtual time. It refers to the scheduled time of the Event that triggered the Application.

Note: You cannot use this term in an Event definition on its own. You can, however, specify a submission time for a job such as NOW PLUS 1 HOUR.

OF

Indicates a connecting word for scheduling terms, for example, LAST DAY OF MONTH or 1ST-9TH WORKDAY OF EACH YEAR.

ON

Indicates a connecting word indicating that a date follows, for example, 7PM ON THURSDAYS instead of 7PM THURSDAYS.

ONCE

Indicates that you want an Event to run once only.

OR

Indicates a connecting word that is used as a conjunction. This term lets you specify either of two similar scheduling terms. For example, 9PM ON FIRST MONDAY OR TUESDAY OF MONTH means that if the month starts on Tuesday, then this job will run on Tuesday. Otherwise, it will run on Monday.

Note: The term OR implies whichever instance occurs first and should only be used when it is meaningful. A statement such as 1ST JANUARY or 2ND FEBRUARY is meaningless to CA WA.

PLUS *n units*

Requests an adjustment by adding a value, which must be a whole number. For example, you can specify PLUS 2 WEEKDAYS, PLUS 0 WORKDAYS, or PLUS 1 WEEK. To obtain the first day after the last workday of the month, you can use LAST WORKDAY OF MONTH PLUS 1 DAY.

PM

Specifies a time between 12:00 and up to, but not including, 24:00. It identifies a preceding number as a time of day, for example, 6PM.

REALNOW

Specifies the actual time that the Event is triggered. For example, you can specify the start time for a job as REALNOW PLUS 1 HOUR.

ROUND

Indicates that the computed time is an integral multiple of the units parameter. You can use ROUND with EVERY n UNITS phrases, for example:

EVERY 1 HOUR ROUND requests every hour at the hour mark.

EVERY 6 HOURS ROUND requests the times 00:00, 06:00, 12:00, and 18:00.

EVERY 5 HOURS ROUND provides the next hour that is an integral multiple of five hours starting at 00:00 on the current day (that is, 00:00 today, 05:00, 10:00, 15:00, 20:00, 1:00 tomorrow, 06:00, and so on).

STARTING

Specifies the starting point for a repeating schedule. You can use an actual date or time after STARTING, for example, DAILY STARTING NOV 6, HOURLY STARTING 2PM.

Note: You cannot use this term in Run and Do not Run statements for jobs and Applications.

THE

Indicates an optional connecting word, for example, 5TH DAY OF THE MONTH instead of 5TH DAY OF MONTH.

THIS

Specifies the current period, for example, 1ST WORKDAY OF THIS MONTH. You do not usually need the term THIS when you schedule jobs; however, it can be useful when you want to ensure CA WA generates date and time variables in the current period. For example, to simulate an Event for the last workday of this month plus one day, use LAST WORKDAY OF THIS MONTH PLUS 1 DAY.

TODAY

Specifies the current virtual or scheduled day.

TOMORROW

Specifies the day after today.

UNTIL

Specifies the ending point for a repeating schedule, for example, DAILY AT 9AM STARTING TOMORROW UNTIL 1JUL2005. This term is identical to the term ENDING.

Note: You cannot use this term in Run and Do not Run statements for jobs and Applications.

WEEK

Specifies a seven-day period. The first day of the week depends on your geographical location.

WEEKDAYS

Specifies Monday through Friday.

WEEKEND

Specifies a grouping of Saturday and Sunday together. For example, ANYDAY OF WEEKEND resolves to Saturday and Sunday.

WEEKLY

Specifies every one week.

WITHIN

Indicates a connecting word for period processing. For example, 5TH MONDAY WITHIN MONTH selects only the 5th Monday in the months that have five Mondays. CA WA ignores the other months.

WORKDAYS

Specifies all of the days worked in each calendar. Workdays exclude holidays.

YEARLY

Specifies every one year.

YESTERDAY

Specifies the day before today.

How to use Virtual Resources

To be written...

What do those Application Properties control?

Contributed by Alan Monument

When you first create a new CA WA DE Application you are presented with the *Application Properties* dialog box...

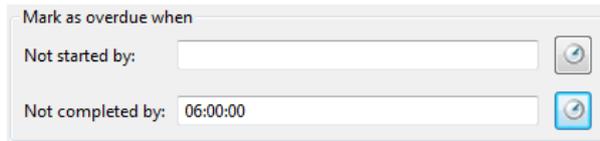
The screenshot shows the 'Basic' tab of the 'Application Properties' dialog box. The 'Name' field is set to 'DEMO'. The 'Runtime name' field is empty. There are several checkboxes: 'Wait for previous generation' (checked), 'Estimate end time' (checked), 'Propagate dueout time' (checked), 'Hold on submission' (unchecked), 'Do not inherit dependencies' (unchecked), 'Do not trigger if active' (unchecked), 'Require reason for job commands' (checked), and 'Suppress notification when no work selected' (unchecked). Below these is a 'Run frequency' table with columns 'Action' and 'When'. There are buttons for 'Add Run', 'Add Do Not Run', 'Delete', and 'Show in Calendar'. The 'Default agent' section has 'Agent' set to 'AGENT' and 'Load balancing' selected. The 'Job wait for previous generation' section has a dropdown menu. The 'Tag' field is empty, and the 'Comments' field contains the text 'App to demonstrate a bunch of dSeries functions.'

This dialog box can also be reached later while maintaining your application by right mouse button clicking in the white space around the job workload objects in the Define perspective and selecting the *Properties* option (hint: you will need to ensure no job workload object is currently selected for this to work).

Wait for previous generation ensures your application runs serially; that is the current *generation* of the application has to finish before the next requested iteration (“*generation*” in CA WA DE terms) is allowed to start. Simply ticking this check box will enable you to trigger multiple iterations of the same application, but only one will run at a time, the rest get queued in the order they were triggered. This makes it easy to trigger several days of an application at once, should there be an issue with the application that has forced it not to be executed for a number of days; each day will then complete in its entirety before CA WA DE allows the next day to run.

Hint: If you know that your application is divided into self-contained components, you might be able to orchestrate the parallel running of subsequent generations of your application by using CA WA DE’s *SubApplication* capability. This capability dramatically reduces the “catch up” time when running multiple days’ work to bring the processing up to date after a major application failure – see the topic *SubApplication: Orchestrating parallel running of applications* on page 62 for details.

Propagate dueout time check box instructs CA WA DE to enable *anticipated end times* and *critical path analysis*, as well as *propagate dueout times* in the Application. Propagating dueout times enables you to receive an early warning that jobs in the Application are running late. All that is then necessary is to *Edit* the time critical job in your application and specify the time that the job would need to complete by; use the *Not completed by* field in the *Time Dependencies* tab of the workload object’s properties as follows...



CA WA DE will then propagate dueout times for all the upstream jobs based on historical runtime data. If any of them should run too long and exceed their dueout time, thus potentially impacting the dueout time for the critical downstream job, the problem job will then go into an overdue status. This in turn could trigger notifications in the problem job to send emails, alerts, and/or SNMP traps.

Hint: Ideally you would set this *Not completed by* time to be some time after the normal average end time for the job, but before the actual SLA/OLA time required by the business. This way you will reduce the number of false alarms due to work taking a little longer than normal, yet still get enough advanced warning that something has potentially gone wrong so that you have time to fix the issue before impacting the SLA/OLA in place with the business.

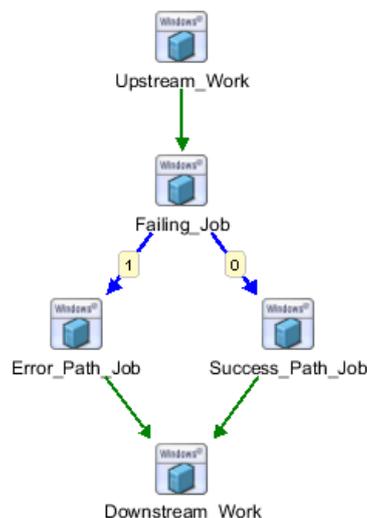
Require reason for job commands check box allows you to force the user to provide a reason for issuing a command against a job. Usually when a user issues a command against a job in the Monitor Perspective, the user can optionally provide a reason for issuing the command. Ticking this check box makes it mandatory to provide a reason for the specific Application.

Do not trigger if active check box allows you to suppress the triggering of an Application that is already running. You might for example use this where an application is being regularly and often triggered, perhaps by transactions arriving via files or data base insertions that then trigger an event. At peak times new events may arrive before the previous event has run its course. This option can be used when the application can cater for the new event in a subsequent triggering of the application, otherwise use the *Wait for previous generation* check box to serialise the running of this generation of the application.

Taking alternative paths through the job stream

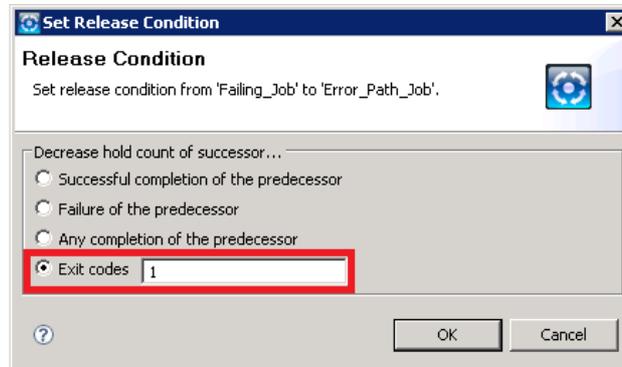
Contributed by Alan Monument

There will be times when you want to take one of two (or more) paths through a job stream. This topic explores one way of accomplishing this where the job stream is to continue when a job is known to fail and an alternative path needs to be followed. The following job will be used as an example...

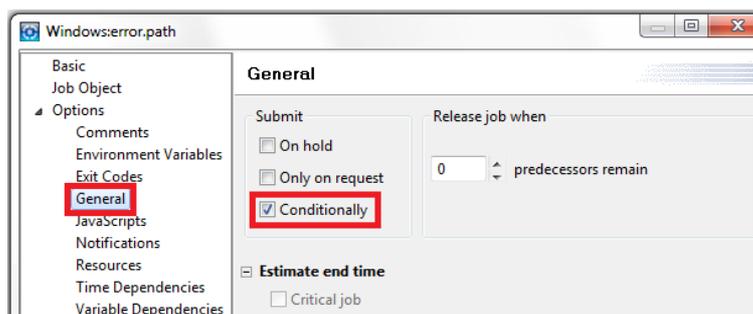


CA Workload Automation DE (dSeries Edition) Cookbook

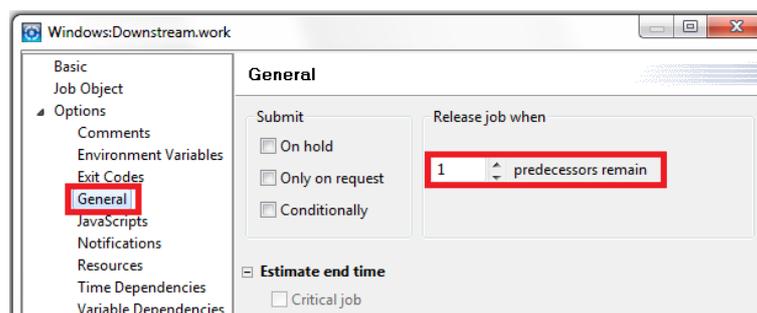
The dependency links between the *Failing_Job* and the successor jobs have been edited (right mouse click on the dependency arrow between the jobs and select *Release Conditions* from the pop-up menu) and appropriate exit codes have been defined. The following example shows the *Release Conditions* for the path to *Error_path_Job*...



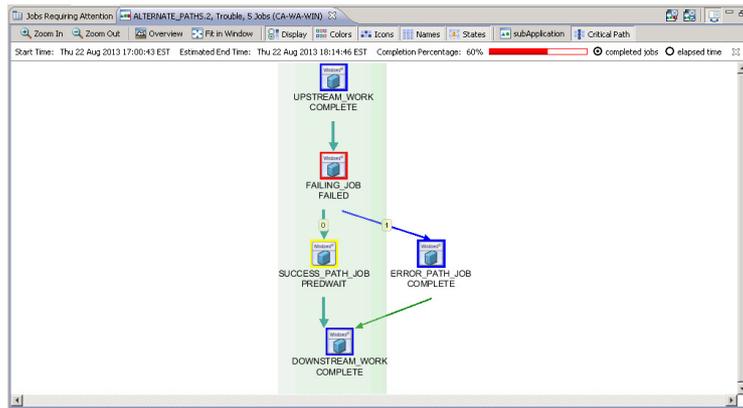
As only one of the two successor jobs will run, they both need to be marked as *Conditional* jobs, so that the job on the path that was not taken will be updated to a BYPASSED status when the application eventually completes, this allows successor jobs to run even though an upstream predecessor has not run successfully. Jobs are marked as conditional by checking the *Submit Conditionally* box in each job's *General* tab...



Finally the *Downstream_Work* job can be instructed to be released when one predecessor remains by editing the job's *General* tab and setting the *Release job when "n" predecessors remain* field as shown below...



When the application runs and goes down the success path, the *downstream_job* will run and the entire application will be marked as COMPLETED and the *error_path_job* will be BYPASSED. If the application takes the error path, the application will run to the end but the *failing_job* will remain in a FAILED state for further operator analysis (as shown below). When the operator changes the *failing_job*'s status to Completed it will allow the entire application to complete and bypass the *success_path_job*.

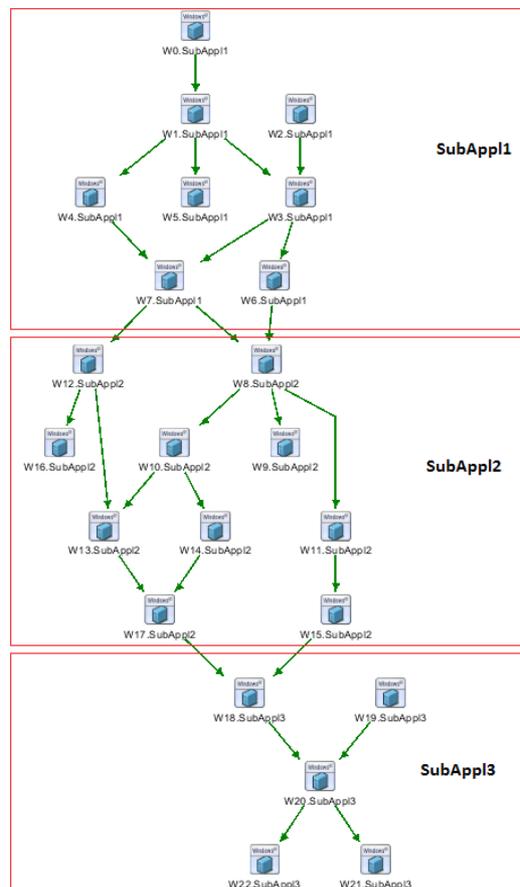


If you want to allow the application to complete automatically with no further operator intervention, it will be necessary to insert an alert notification in the failing_job that in turn runs a Javascript that marks the FAILED job as COMPLETE.

SubApplication: Orchestrating parallel running of applications

Contributed by Alan Monument

Sometimes you may have some processes that are frequently triggered and need to be run as quickly as possible; however waiting for the previous generation of the application to complete before the next can start may introduce un-necessary delays and latency into the processes that you cannot afford to live with. Where you know that your application can be divided into self-contained “sub-applications”, then you will be able to leverage CA WA DE’s very powerful *SubApplication* capability to parallel stream your application’s execution.



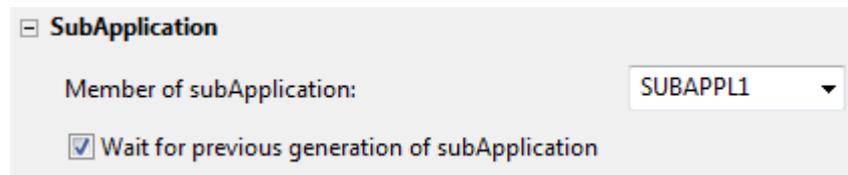
CA Workload Automation DE (dSeries Edition) Cookbook

To do this your application needs to satisfy certain criteria:

- It needs to contain jobs similar to those shown in the first red box named SubApp1 (above) that can be grouped together into a self-contained *sub-application*, which is required to run in isolation from the next generation of the Application
- once the last job in this *sub-application* has completed, the next generation (i.e. the next day's processing) of those same jobs can then run again without impacting any of the rest of the Application's remaining jobs (i.e. those jobs in SubApp2 and SubApp3)
- The more groups you can identify that are self-contained, the more generations of the Application you will be able to parallel run

Once jobs groups are identified, you can assign an arbitrary name to the group, as has been done to the jobs in the job flow diagram above; they have been grouped into SubApplications: SubApp1, SubApp2, and SubApp3. You can use any name you like.

In the Desktop Client, once you have created your Application and associated objects, you can then *Edit* each workload object and in the General tab properties you will find a SubApplication section. Simply enter the allocated *SubApplication* name. Once you have done this for the first time you will be able to quickly select that same text as it will appear in subsequent workload objects in a drop down menu; as in the example below. Notice how the *Wait for previous generation of subApplication* check box has been checked – this will force subsequent generations of these same jobs to wait until the entire SubApplication group has completed, before the next generation of jobs in that SubApplication are allowed to run.

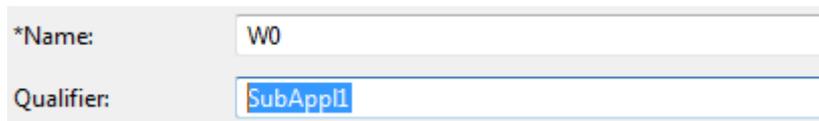


SubApplication

Member of subApplication: SUBAPPL1

Wait for previous generation of subApplication

Hint: To help graphically identify what job belongs to what SubApplication group, you can use the optional job name *Qualifier* to specify the name of the SubApplication that the job resides within – see the earlier graphical representation of this application to see how the workload object names get generated.



*Name: W0

Qualifier: SubApp1

Important: When using SubApplications you will need to uncheck the *Wait for previous generation* check box in the Application's Properties to allow multiple generations of the same Application to be eligible to run at the same time. The above *Wait for previous generation of subApplication* check box in each job will then control what generation of jobs are allowed to run within each of the parallel running Applications.

Once the above has been done, multiple generations of the same Application can be triggered. In the example above, up to 3 generations of the same application could be running in parallel, as the application has 3 SubApplications. As an example of how this work would flow; if we were to trigger 3 generations (G1 through G3) of the above Application:

- when G1 SubApp1 starts running, G2 & G3 SubApp1 would be waiting on the previous generation (G1)

Managing/Monitoring Workloads by Exception

Contributed by Alan Monument

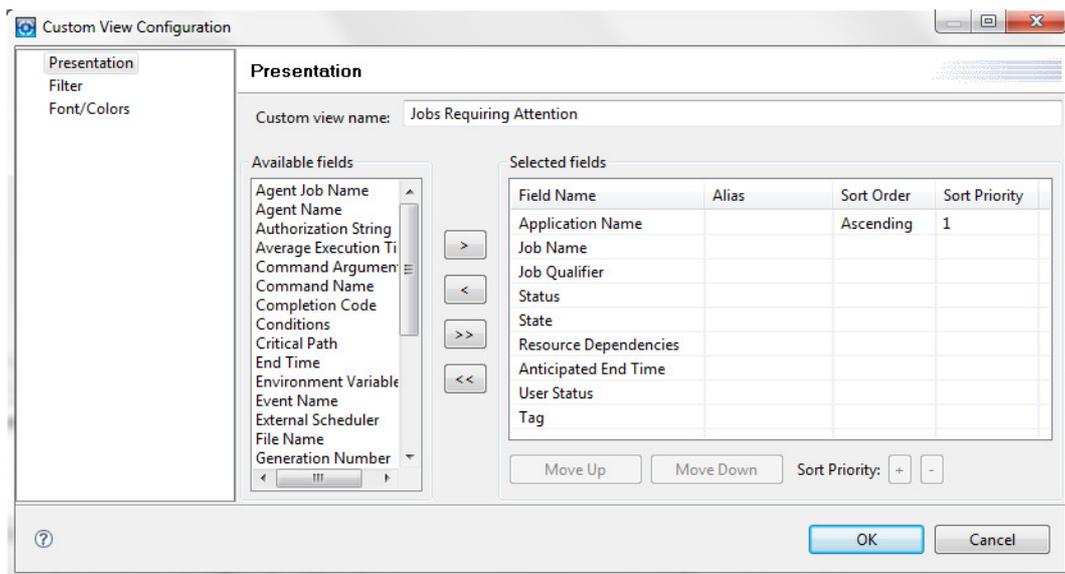
The best way to monitor and manage workloads in CA WA DE is to have visibility of and to manage only the problematic work; and to ignore all the work that is running correctly. This topic explains how to set up views in the Monitor perspective to only see those jobs that need attention.

Click on the Monitor perspective menu button and clear all the currently open panes on the right of the screen by clicking on the X in all the Tabs until all the panes have been cleared.

Creating your own personalised Custom View

Click on the Custom Views tab at the top of the left hand pane, and then double click on the following items in the list to open up monitoring panes for: *Waiting, Held, Ready, and Completed* jobs.

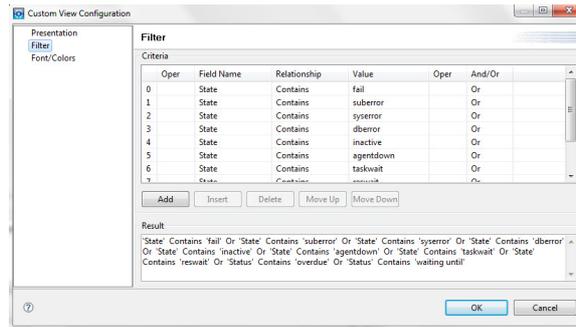
In addition to using the provided “canned” Views, you can also create your own Custom Views. Simply click on the Create a New Custom View icon  at the top of the left-hand Custom Views pane, and complete the resultant dialog window by giving the new Custom View a name, we will use the name *Jobs Requiring Attention* in this case, and select appropriate fields you want to see displayed by clicking on the required ones in the Available Fields column and clicking the right arrow > button to move those fields to the Selected Fields column in the right-hand table as follows:



The Selected Fields column decides what is going to be displayed in our new custom view. These Selected Fields are a useful starting point; however go ahead and choose different fields or add additional ones if you wish – after all this is your Custom View. You can also come back later and change this selection to satisfy your changing requirements.

Now you will need to provide some rules around what jobs you want to display in this view. This is accomplished by clicking on the *Filter* option in the hierarchy tree in the left-hand pane of the custom View Configuration window. To add filter items, click on the Add button and complete the fields in the Filter Criteria table by selecting items off the drop down menus and entering values. Keep pressing the Add button to create additional rules and complete the dialog as shown below (add additional ones or experiment if you wish):

CA Workload Automation DE (dSeries Edition) Cookbook



Click OK to save your new Custom View definition and double click on the *Jobs In Trouble* view to open that view.

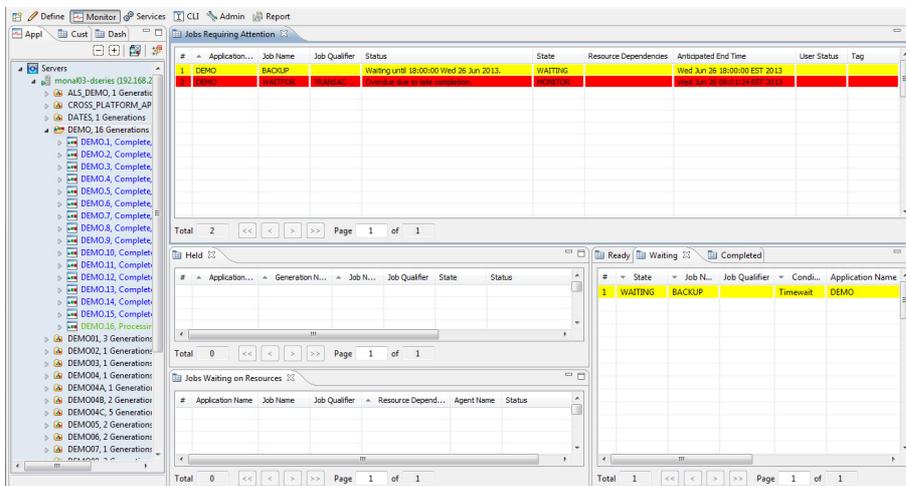
You can now click on each of these new tabs in the right-side of the window to see the contents of the various panes. It would be nice however to be able to see all these panes at the same time. The good news is you can!! Click and drag the *Held* tab towards the bottom of the window. Keep moving the mouse down the screen until you see a ghost window outline creating a new pane in the bottom half of the window. Release the mouse button and the *Held* tab now appears in the bottom half of the window and the rest are in the top half.

Now drag the *Waiting* tab to the bottom half of the window and then move it to the right until the ghost window divides the *Held* pane into two. Release the mouse button and the *Waiting* tab now appears in the bottom right half of the window, the *Held* tab is on the left, and all the remainder are in the top half.

Now drag the *Ready* tab to the bottom right over the top of the *Waiting* tab pane until a ghost window appears around the *Waiting* pane. Release the mouse button and the *Ready* tab now appears in a next to the *Waiting* tab. You can view the contents of either tab simply by clicking on the tab.

If required, you can size each set of panes by running the mouse pointer over the dividing margins until it turns into a double ended set of arrows, then you can then click and drag to move the window's borders.

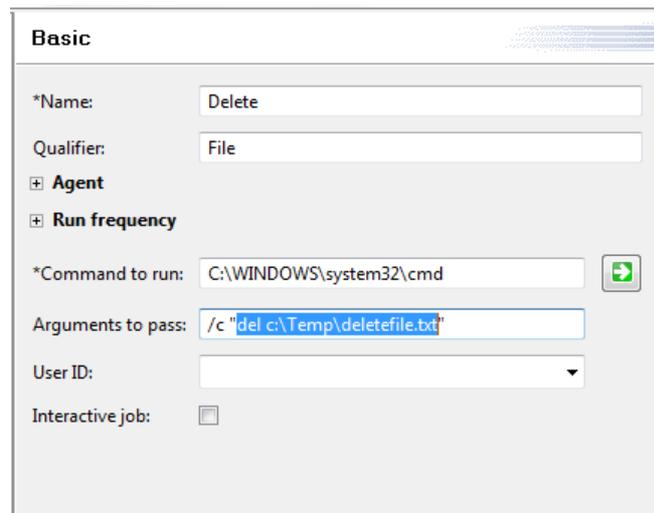
Now click on the *Application* tab in the far left pane as shown below and your screen should look something similar to the example below, which also contains a couple of extra tabs: *Completed* and *Jobs Waiting on Resources* (this last one is another user created custom view). The idea is you create and use customer views that satisfy your monitoring requirements. The most important tab below is the *Jobs Requiring Attention* view, as this will be displaying those jobs in trouble and in most need of operator attention. The other panes allow for selective viewing of other work.



Running a Windows Command Prompt command

Contributed by Alan Monument

Sometimes you will want to run a Windows operating system command that would normally be run from a Windows Command Prompt window. To run these type of commands you will need to run the Windows command prompt processor and specify the command that you want to execute as an argument as shown in the following example...



The screenshot shows a configuration window titled "Basic" with the following fields and options:

- *Name:** Delete
- Qualifier:** File
- Agent:** (Expanded)
- Run frequency:** (Expanded)
- *Command to run:** C:\WINDOWS\system32\cmd
- Arguments to pass:** /c "del c:\Temp\deletefile.txt"
- User ID:** (Dropdown menu)
- Interactive job:**

Note that the `/C` argument in the *Arguments to pass* field tells the command processor to execute the following command and then terminate the command processor session so that the CA WA DE job will then complete; otherwise the command processor will run indefinitely and the job will never complete. Place the actual command you want to run between quotes if it contains any spaces, such as in the example above. If required, multiple commands can be invoked using this same method by using a *command separator* between the commands – for more details about structuring these arguments run a Windows command prompt and then execute `help cmd`.

Future Considerations note from the CA Support page:

CA Technologies is committed to maintain support for released features in subsequent releases of our products where possible. Unfortunately, sometimes operating system and application vendors change their products in ways that can have a negative impact on CA products.

For example, changes Microsoft made to Session 0 in Windows impact support for interactive applications. In the document at <http://msdn.microsoft.com/en-us/windows/hardware/gg463353.aspx>, Microsoft stated the following regarding Session 0:

In Windows XP, Windows Server 2003, and earlier versions of Windows, all services run in Session 0 along with applications. This situation poses a security risk. In Windows Vista, Windows Server 2008, and later versions of Windows, the operating system isolates services in Session 0 and runs applications in other sessions, so services are protected from attacks that originate in application code.

In addition, starting with Windows 8 and Windows Server 2012, Microsoft disabled the Interactive Services Detection Service. The Interactive Services Detection Service is required for applications running in Session 0 to interact with users on the system. To enable this service, the administrator must edit the registry. If Microsoft

removes this service from a future version of Windows, support for interactive jobs running in Session 0 will no longer be possible.

Many customers run interactive applications as part of workloads on Windows. Customers using an older Windows system can interact with these applications because both the applications and the logged-in user run in Session 0. Jobs can also run in user sessions, but there are some limitations. To share the session, the interactive applications must run as the user. Also, to direct the application to the proper session, the agent requires a single active session per user. System resources also have specific limits in Session 0 that could be a problem if too many jobs are running simultaneously.

Many customers using the CA AutoSys Remote Agent or the CA Unicenter Universal Job Management Agent rely on Session 0 support to run interactive jobs. To assist these customers migrating to CA Workload Automation Agent for Windows, CA has added this support in r11.3 SP1 Cumulative fix 3.

CA will continue to support interactive applications in Session 0 on Windows operating systems that support the behaviour. However, since Microsoft no longer actively supports this behaviour as of Windows Vista and Windows Server 2008, CA can only offer best-effort support on those and later versions of Windows. By continuing to run interactive jobs in session 0 on those platforms, customers may incur security and stability risks.

Because CA knows that interactive Windows jobs are important to many customers, CA is working closely with Microsoft to determine an appropriate long-term solution. While CA strives to avoid or minimize any required changes to job definitions or business processes, some changes may be necessary. CA hopes to include the long-term solution in an upcoming service pack.

If you have any questions or concerns, or would like to work with CA and Microsoft to develop the appropriate long-term solution, please contact your CA support representative.

Running a Job to Interact with a Windows Application

Contributed by Rio Satriyo

Sometimes you might need to automate the running of an interactive application that is part of a larger job stream with job dependencies. For example you may want to perform the following steps that would traditionally be done by an operator during end of day processing:

1. click the Windows Start button
2. Choose hmailServer Administrator
3. Fill out the password
4. Click the Exit button

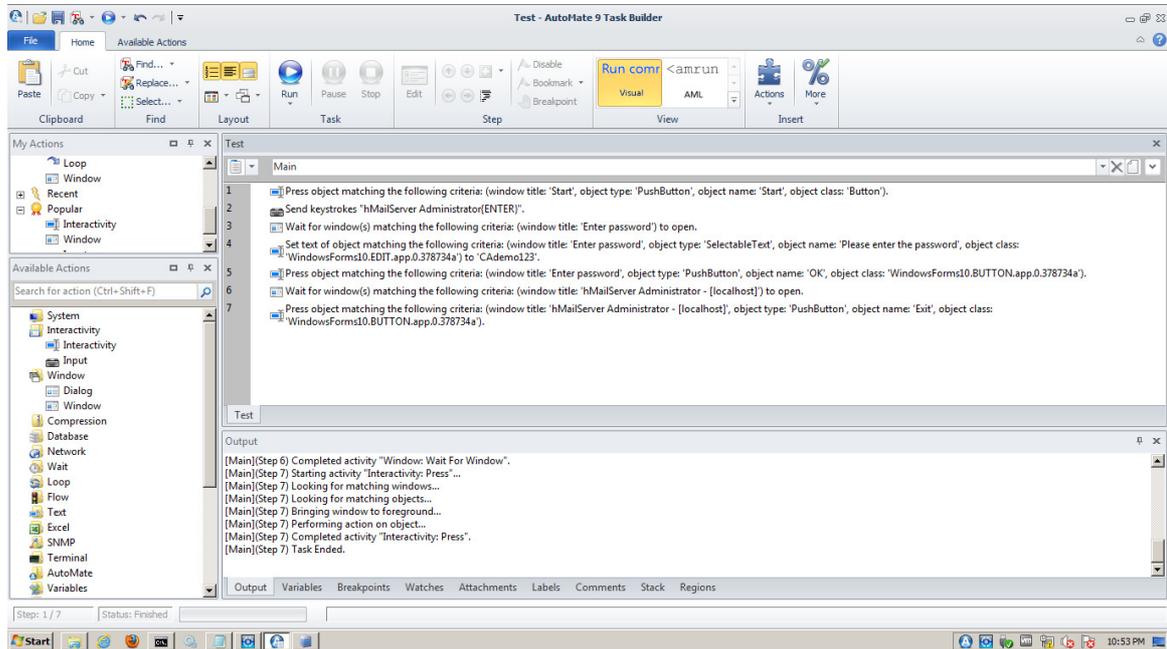
This can be accomplished in CA WA DE by using a third party GUI automation tool such as AutoMate 9, Auto IT, HP Load Runner, etc. In this example we will use the Task Builder component of AutoMate 9 from Network Automation (www.networkautomation.com/automate/9). Automate has hundreds of activities that can be dragged and dropped from a palette into an ordered sequence for execution later. These activities can include decision making, looping, variable and array manipulation, and many other tasks to completely automate an operation.

Build an automation script in AutoMate:

1. Open Automate 9 Task Builder

2. Create Steps in the Automate9 Task Builder

- ✓ Automate 9 has many action features that can be used to record the user interactive session



3. Save the activities as a task in a suitable folder (File → Save As) - provide an appropriate name such as **Testing.Am1** for the file.

Add a Workload Object to run the automation

1. Go to the CA WA DE Desktop Client
2. In the Define perspective, download the application into which you want to add the new automation step
3. From the System Palette, drag a new Windows workload object into your application's canvas
4. Establish dependencies to/from the new workload object
5. Double-click/Edit the new workload object:

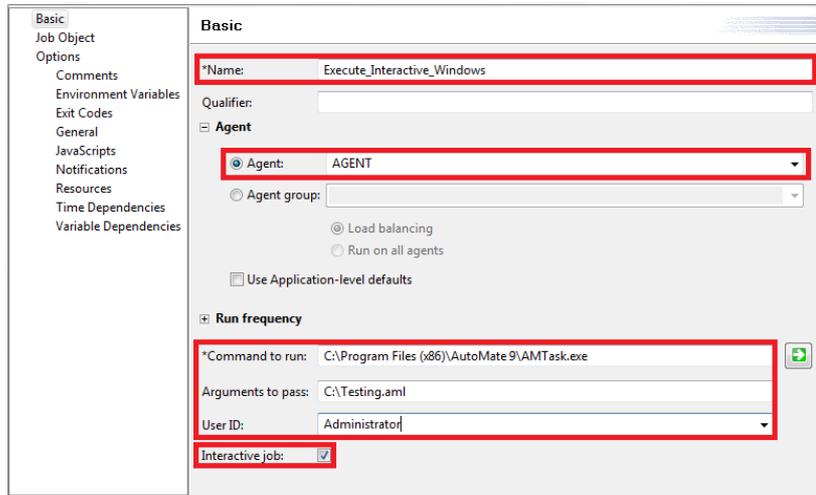
- a. Give the new workload object a Name
- b. Select the system Agent that is running on the server where you saved the automation script (above)
- c. In the **Command to run* parameter, type the following (or similar depending on where the AMTask.exe resides):

C:\Program Files(x86)\AutoMate 9\AMTask.exe

- d. In the *Arguments to pass* parameter, type the following (or similar depending on where the activities script was saved):

C:\Testing.am1

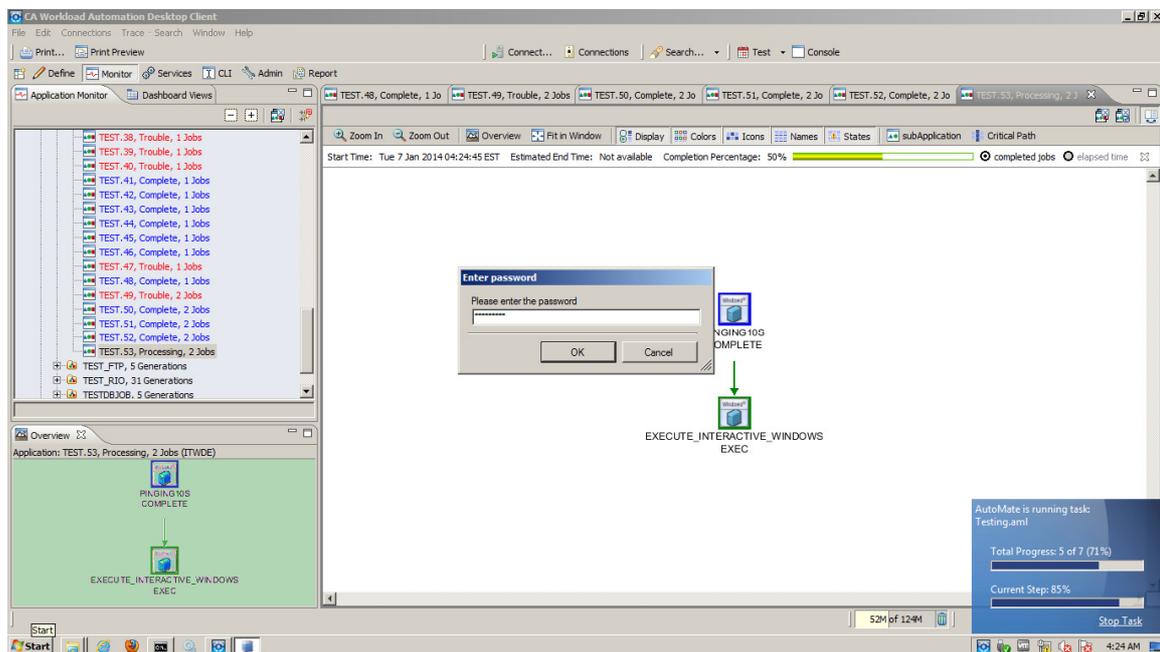
- e. Set the *User ID* with an appropriate user
- f. Check the *Interactive Job* checkbox.



Note: With a *User ID* and the *Interactive Job* checkbox checked, you will be able to see the all steps that you recorded before, executing when the job runs

Test the automation

1. Using the CA WADE Desktop Client, run the application containing the new workload object....

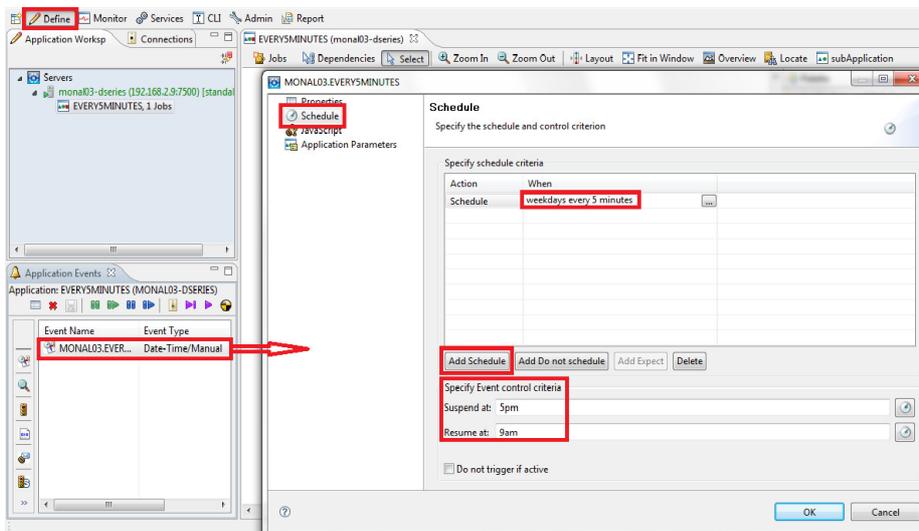


2. The new workload object and associated activity script runs automatically. The entire business process eventually completes when the automation job and associated successor jobs all complete.

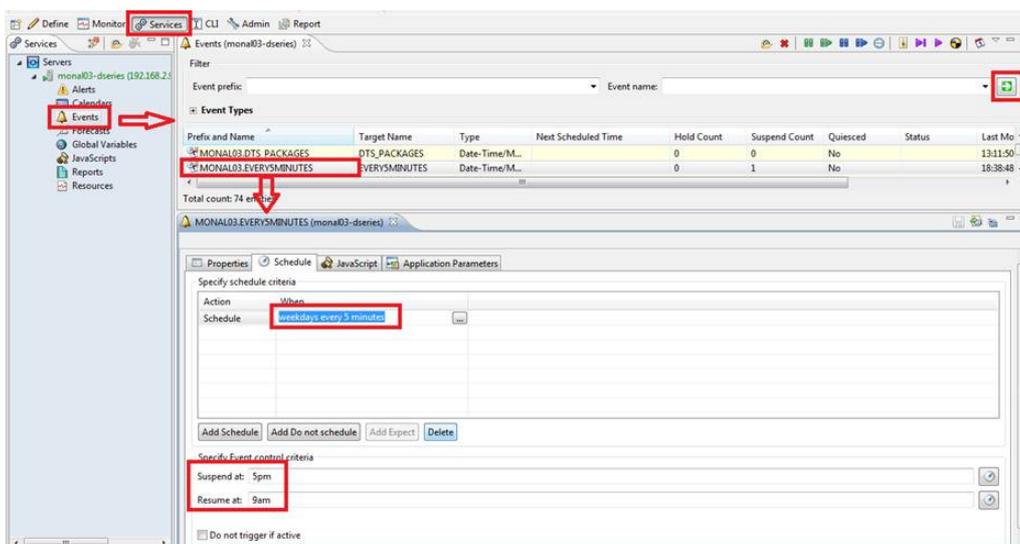
Running an Application every “n” minutes or hours

Contributed by Alan Monument

If you need to run an application every “n” minutes, or hours, you can accomplish this by creating your application in the *Define* perspective and then editing the associated event; either by right mouse clicking and selecting Edit from the pop up menu, or by double clicking on the event name (see below). Using the resultant Event dialog, click on the *Schedule* tab, click the **Add Schedule** button, and then enter your run criteria into the *When* column (daily or **weekdays every 5 minutes** for example). You might also want to specify a time period in the *Specify Event control criteria* fields to sets limits on when the application will run like this – in the example below the application will run between 9am and 5pm. Also consider whether you want to trigger the application within your specified run times if the previously triggered generation is still running – set the *Do not trigger if active* check box appropriately.



Once the application is *uploaded*, CA WA DE will then trigger this application every 5 minutes between the hours of 9am and 5pm. Note if you have already created an event, you can also modify the event schedule parameters from the *Services* perspective -> *Events* tab.



Skipping non-essential workloads during Disaster days

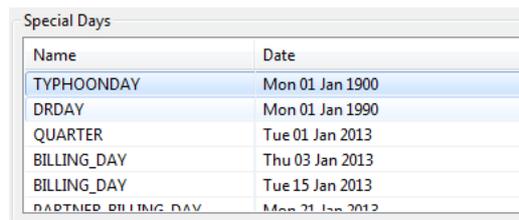
Contributed by Alan Monument

Sometimes there are days where you do not have full processing capacity due to a data centre or network outage, or you have a “shrunk” processing window due to some form of major application disaster or perhaps a catastrophic weather condition. These things usually only happen when you least expect them and you are not in the best position to effectively deal with them. They are known to most of us as “Disaster Recovery” or “Typhon” days.

On these days you will need to quickly identify the critical from the non-critical jobs, and then modify your job streams to only run those really critical workloads. This takes time and is prone to error when under the duress of these kinds of situations. Fortunately with a little bit of planning and time on your part, CA WA DE has powerful capabilities that can help you prepare to automatically handle these days when they do occur.

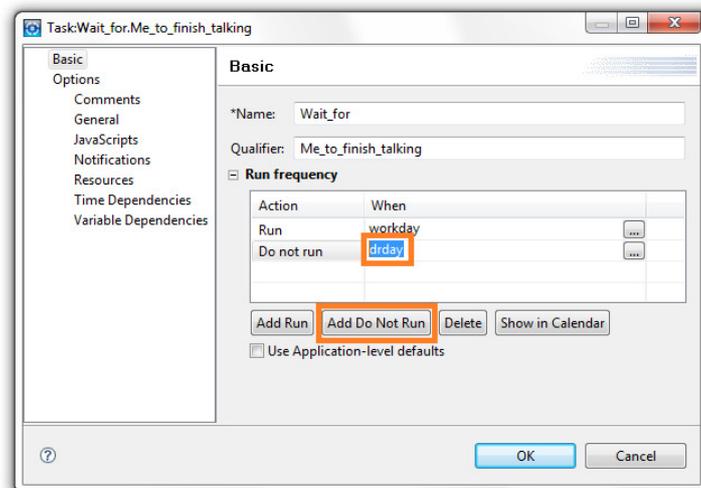
A combination of two of CA WA DE capabilities allows you to automatically skip the running of non-critical workload when a disaster occurs. These are the CA WA DE *Calendar* and the *Run Frequency* section of a workload object job definition.

First the Calender; in the System Calendar define either a DRDAY and/or TYPHONDAY and set the date to Mon 01 Jan 1990 so that it will have no impact on any of the currently running workloads...



Name	Date
TYPHOONDAY	Mon 01 Jan 1990
DRDAY	Mon 01 Jan 1990
QUARTER	Tue 01 Jan 2013
BILLING_DAY	Thu 03 Jan 2013
BILLING_DAY	Tue 15 Jan 2013
PARTNER BILLING DAY	Mon 21 Jan 2013

Now identify those workloads that you do not want to run when the situation demands it. Be aware of data dependencies between jobs – this may take a bit of analysis by the application owners, but it is well worth the investment up front as you will be able to respond quickly and with confidence when a disaster strikes – and it will!! Now edit all those workloads that you can safely skip, expand the *Run Frequency* section of the *Basic* tab and click on the **Add Do Not Run** button. Enter **DRDAY** (or whatever special day name you have chosen to deal with your nightmare day) into the *When* column of the new entry and save the change by clicking on the **OK** button...



Once completed, you will now have your insurance policy in place and can rest easily knowing that you can react quickly to the day when everything starts to go wrong. When that day does come; you simply change the date of DRDAY to match today's date and sit back and watch any newly triggered workloads skip their non-critical processes for the rest of the day.

If you anticipate the disaster day will exceed 24 hours; start creating additional DRDAY entries for the subsequent days. You can always reset the initial DRDAY entry back to its original setting and delete the additional entries as soon as you recover from your disaster situation; and subsequently triggered workloads will resume normal processing. Note that you will need to leave at least one DRDAY definition in the calendar.

Hint: Use Simulate to test that your job flows handle DRDAY correctly by entering **DRDAY** into the *Schedule criteria* Simulate option field. **IMPORTANT:** DO NOT change DRDAY to be a current date as this has the potential to have a disastrous effect on any other running workloads that also use DO NOT RUN on DRDAY.

Using FTP Workload Objects

Contributed by Alan Monument

CA WADE workload objects can be used to transfer files between servers. To accomplish this you first need to get your FTP nodes configured correctly. There needs to be an FTP "server" at one end of the proposed transfer link and an FTP "client" at the other end. The ESP System Agents can be configured with or without FTP capabilities, and can act as either as an FTP client or server.

The CA WADE system agent on the server where you want to issue FTP transfer commands from, will need to be enabled as an *FTP client*. If FTP was not enabled on the agent at installation time, you can still accomplish this by editing the agent's *agentparm.txt* file (located in the agent's installation directory) and restarting the agent.

Next you will need an *FTP server* at the other end of the transfer link for the client to communicate with. If you can already manually FTP files to/from the target server, then the target server will already have an active FTP Server, and you should be good to go too. If not, you will need to configure an FTP server on that server, define users and passwords to allow access to it, etc.

Setting up an FTP User in CA WADE

In the CA WADE Desktop Client you will need to associate an *FTP User* with the *FTP client* node's System Agent (this System Agent is the one located on the server where you will be initiating FTP transfers). Users defined on this agent also store their associated *Password* which are needed to authenticate that user on the FTP server at the other end of the connection. For security reasons, these passwords are stored in encrypted form and cannot be viewed by anybody defining the FTP jobs.

This is a one-time activity, and once established, you can re-use these User definitions in other FTP activities on this agent. You will most likely need to define a number of FTP users with different passwords to access different target FTP servers.

To associate a User Id with an agent you first need to be connected to the CA WADE repository using the Desktop Client as a user with appropriate administrator privileges to modify agent definitions in the Topology browser:

- Click on the *Admin* button in the Desktop Client's menu area, and then double click the *Topology* item under the connection in the left hand pane

CA Workload Automation DE (dSeries Edition) Cookbook

- Double click on the FTP client agent's entry in the *Topology* pane to open the agent properties editor.
- Now click on the *Users* tab followed by the *Add User...* button
- Enter a User to match the target FTP Server's *User Id* and its associated *password*, followed by **OK** to save the entries, and then click on the floppy disk icon to save the agent properties

All of the above is usually a one-time activity which will then enable you to use FTP workload objects in CAWADE applications.

Defining an FTP workload object

To define an FTP workload object:

1. Drag and drop an FTP object onto your application's palette and establish appropriate job dependencies with other workload objects.
2. Double click the FTP object and enter the appropriate parameters:

The screenshot shows the 'Run frequency' dialog box with the following fields and options:

- Job class: [Empty text box]
- *User: [Dropdown menu with 'anonymous' selected]
- *Server address: [Text box with 'ftp-test.mozilla.org']
- Server port: [Empty text box]
- *Remote file name: [Text box with 'pub/README']
- *Local file name: [Text box with 'C:\temp\README.txt']
- FTP options**
 - Transfer direction: [Radio buttons for 'Download' (selected) and 'Upload']
 - Transfer code type: [Radio buttons for 'Binary', 'ASCII' (selected), and 'EBCDIC']
 - Local user: [Empty text box]
 - Compression level (0-9): [Spin box]
 - SSL Connection: [Dropdown menu with 'False' selected]
- FTP site commands**
 - Table with columns 'Command' and an empty row.
 - [Add] button
 - [Remove] button

In this example:

- a. The FTP workload object will use a User named **anonymous** – this user would be defined in the associated agent's *User* definition as described earlier
- b. it is going to download the file **README** from the target FTP Server
- c. from the **/pub** directory
- d. on the **ftp-test.mozilla.org** FTP Server
- e. and save it as a local file named **README.txt**
- f. in **ASCII** file format
- g. under the **c:\temp** directory

3. Click **OK** to save

Notes:

- If for any reason your FTP workload object returns a **SUBERROR** status at runtime; double click on the object in the *Monitor* perspective and check the *Details* dialog box for potential clues to the problem. Also double check the parameters you entered into the job definition for correctness.
- Sometimes you might encounter a situation where an FTP workload object process successfully signs on to a target FTP server but then seems to hang and eventually fails with a "**Download failed: PORT unsuccessful**" message in the FTP job's spool file. By default an agent's FTP capability works in *active FTP mode* and uses outbound port 21 to establish a connection, but then uses a second port established by the FTP server (usually port 22) to perform the data transfer. If you experience the above message, try placing an **ftp.passive=true** statement into the agent's `agentparm.txt` parameter file to force the agent to run in *passive FTP mode* and only use port 21. Note that any change to the agent's `agentparm.txt` parameter file will require a restart of the agent service to activate that change.

Delaying job execution by less than a minute

By Nitin Pande

In CA Workload Automation DE, users can define a delay in submission in the job definitions. This delay has to be in minute increments and cannot be defined in seconds. However, there may be situations where a business requirement needs to delay submission of a job by a few seconds.

This can be achieved by inserting a time job before the job that needs to be delayed. The inserted job can run a sleep or pause command. In Unix/Linux systems sleep can be defined this way:

```
sleep n
```

This will suspend the execution of the following job by *n* seconds, where *n* is a numeric value. The following example will wait 5 seconds: `sleep 5`

In Windows, there is no built-in equivalent of 'sleep'. Users can use the following command to achieve similar results.

```
ping 1.1.0.0 -n 1 -w n > null
```

Set the ping to an un-reachable host and set the value *n* after the `-w` flag as the time to wait. Increment the `-w` parameter in milliseconds (for example 1000 ms = 1 sec). The following example will wait 5 seconds:

```
ping 1.1.0.0 -n 1 -w 5000 > null
```

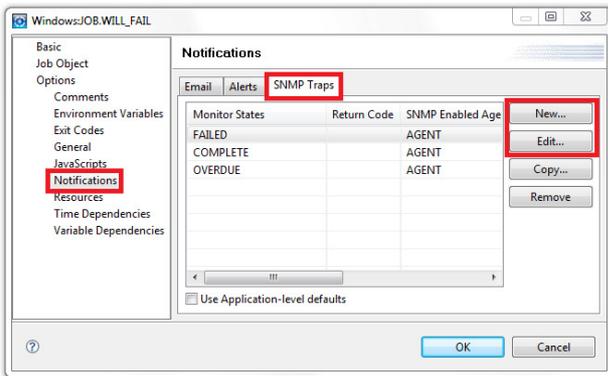
Using the Desktop Client's SNMP Messages Viewer

Contributed by Alan Monument

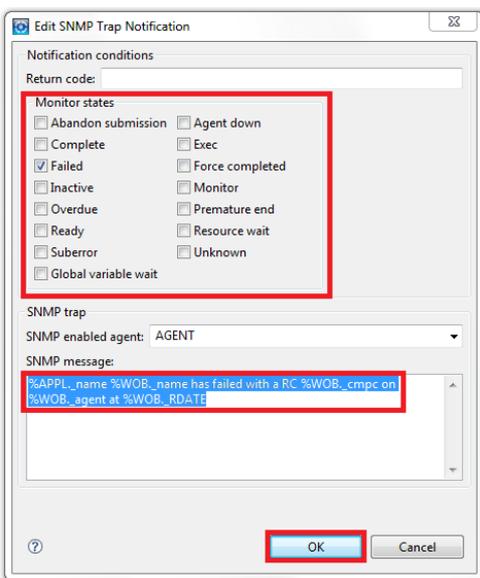
To ensure you are structuring SNMP notification messages correctly in your job definitions, you can use the Desktop Client's built-in SNMP Messages Viewer to trap and display generate SNMP messages right on your desktop. The following topic discusses how you accomplish this..

Define the structure of your SNMP message

In the *Notifications* tab of a workload object, create a *New...* or *Edit...* an existing *SNMP Trap* Notification....

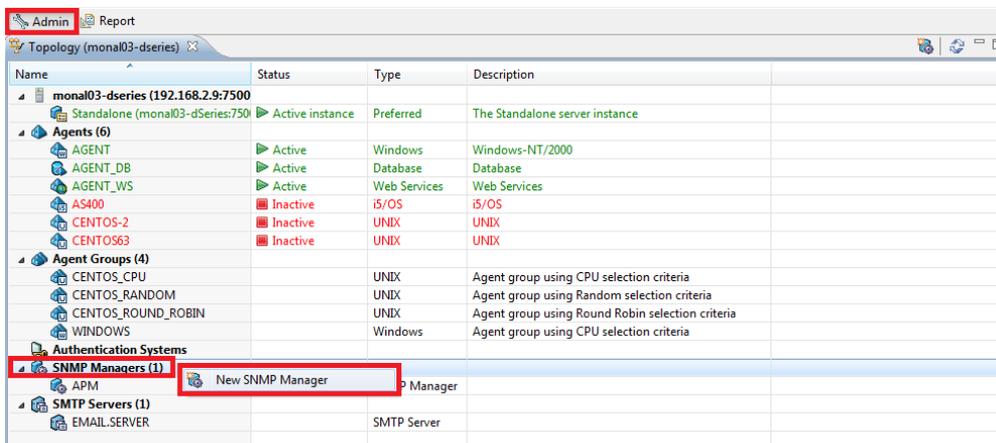


In the resultant *SNMP Trap Notification* dialog window, select the desired *Monitor states* that will generate the SNMP message, and then define the message using a combination of fixed text plus any symbolic variables described in the Programming Guide, as shown in the following example...



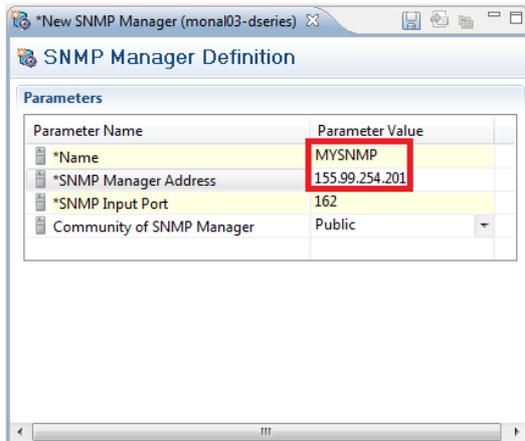
Setup a temporary SNMP manager on your workstation

Before testing the job, go to the Desktop Client's *Admin* perspective, double click the *Topology* tab and then right mouse click on the *SNMP Managers* line in the Topology window, and select *New SNMP Manager* from the pop up menu to create a temporary SNMP manager on your workstation...



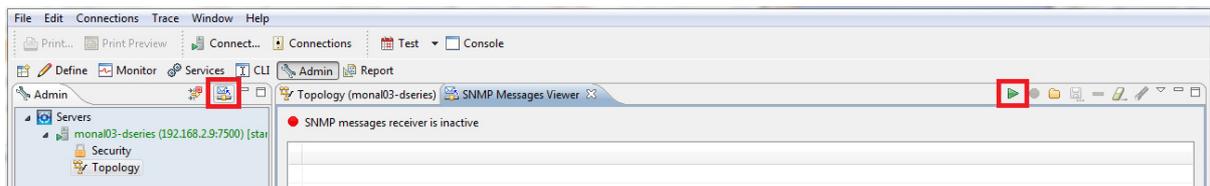
CA Workload Automation DE (dSeries Edition) Cookbook

Determine the IP address of the computer from where you are running the Desktop Client and enter that into the new SNMP Manager Definition window along with a name for the SNMP viewer (this can be any name)...



If necessary change the SNMP Input Port and Community values before clicking on the diskette icon to save this new definition.

While still in the *Topology* browser, click on the *Open SNMP Messages Viewer* icon at the top of the *Admin* tab and click on the *Play* icon (or press Ctrl+R) to *Start SNMP Messages Receiver* in the resultant SNMP Messages Viewer window...



Specify the Port number in the resultant *Start SNMP messages receiver dialog and set port dialog* if you changed the default port number above, and click **OK**. You should now see the status change in the top of the window indicating that *SNMP messages receiver is running on port: nnn*.

Run jobs to generate SNMP messages

Now run you application and force the job(s) to invoke their error conditions.

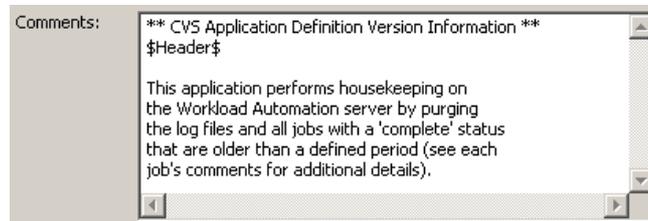
Hint: It may be simpler to duplicate an SNMP Trap Notification you want to test and set the *Monitor state* to *COMPLETE* so that the job generates the message without you having to force the job to fail. **Don't forget to remove this later** once you have validated the message is being generated correctly.

In the SNMP Messages Viewer window you should see the SNMP messages generated by your application. Scroll the viewer window to the right to the message you have structured to ensure you are getting what is desired...

CA Workload Automation DE (dSeries Edition) Cookbook

version control system at the same revision level as the rest of the business application to enable all components to be “released” as a group, and to allow for easier auditing.

Workload Automation application definitions can make use of the above mentioned keyword substitution to embed versioning information into the application by placing desired keywords into the application properties Comments field as shown here:



There are a variety of keywords that can be used with CVS and these are all documented in the Keyword Substitution chapter of the CVS manual, which can be downloaded from <http://ximbiot.com/cvs/manual/>.

Exporting the above application definition would then result in the following XML extract for the Comments section. Note the CVS **\$Header\$** keyword in the comments section.

```
<app:appl xmlns:app="http://dto.cybermation.com/application"
name="SERVER_CLEANUP"><app:version>11.1</app:version><app:comment>** CVS Application
Definition Version Information **
$Header$
```

```
This application performs housekeeping on
the Workload Automation server by purging
the log files and all jobs with a 'complete' status
that are older than a defined period (see each
job's comments for additional details). </app:comment>
```

When this source code is checked into the software version control system and checked back out again, the **\$Header\$** keyword is updated with the appropriate versioning information as shown below.

```
<app:appl xmlns:app="http://dto.cybermation.com/application"
name="SERVER_CLEANUP"><app:version>11.1</app:version><app:comment>** CVS Application
Definition Version Information **
$Header: C:\CVSRoot\WA\SERVER_CLEANUP.xml,v 1.1.1.1 2010/10/07 06:00:41 Administrator Exp
$
```

```
This application performs housekeeping on
the Workload Automation server by purging
the log files and all jobs with a 'complete' status
that are older than a defined period (see each
job's comments for additional details). </app:comment>
```

In environments where there are Production and Development instances of Workload Automation servers, the first XML example above would have been exported from the Development instance and checked into the software version control system. Later the same application definition would be checked out (resulting in the second XML example above) and promoted into production by importing the XML into the Production Workload Automation instance.

CA Workload Automation DE (dSeries Edition) Cookbook

the key stores. The *public* key is freely shared with any other nodes that needs to share data with that computer; while the *private* one needs to be kept very secure within your own organisation.

In the example to the right, the external organisation has been sent a *public* key in a file named *alan.monument.pk* that was exported from Computer B (we'll see how we export those keys later) and we can see how the `--import` command is used to import that public key into Computer A's key store.

GPG tells you it has imported a *public key* with a name of Alan Monument (in this instance) and an e-mail of alan.monument@ca.com.

```
C:\Program Files\GNU\GnuPG>gpg --import alan.monument.pk
gpg: key C9C1D837: public key "Alan Monument (Alan's CA Work e-Mail) <alan.monument@ca.com>" imported
gpg: Total number processed: 1
gpg:      imported: 1

C:\Program Files\GNU\GnuPG>gpg --armor --export dSeries@ca.com
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.10 (MingW32)

mQMuBEwM7IRG6Crc0hN222IQAB8/Nn3DF8MoX/8a9SjFuz/c0InWuFLES6BhgG
fM2eJ8MZ1//adu5193v2a9pp/a0Gc9qc2hg5x3Nu7ieZMhza0nGnm6RoR8/BL
x7a0r6zngv5Gn4hsMudoGu8DEOnnHk+NDHz0UM/X+MCS8Cfhtvgpf1EAgvGR3NA
ttBo0rYJjiifIoCHEtLk3YchKqz8bTnruURyOfHCAcXTD0JwyITi5DUKnUsIPu88
SeM81cY0sFsnGaiWNEPUsaIU191glJj1DA17+m/+yPb4CyZL4hkKLG4hCPVUtr
p5uz/Z5R4gSMFq1s1a0LcR7vzBDE1zIM4LH78QcPqY06SytQ2s3uMeNda2WYzXc
pUoZuKsMOBjmsG4jZvz01i3duK9d880Qhve9caFxd6ht3EPmkaRtv/Cxv0v/Ro
UGfvnmndmP8Nf133x+3S1tUftP3UveGVB4/U+3KtDhnJFT1i8NqQbVr7HQhafLdJ
Ee4x0Zq4U9kuEu500/5biLoVJLJBHhU+7pBN1J1uW4YsBEW1gasUe0QA0rBz4
ROBBuEtSLdCvLqW0AyuANJRoOBVFEJiUtuUHTKhggDuNa10BLTE8XD1Uth3cNG
DTP3Pa17UCxP8FdzLum6am7Bs00v5cn9R182oPv7dPFSs1b3Bv0/4+nMdyJ0JD
y8NnMz2pF2yU11hi24qsmIn994H+JHc6Nmb1J3q/Zhe1i33UDMecurEG5Dw
4M2Hzq5hZDNV0+1XH0Dht0164bc6eZ1ixncdZmatt12GtQ0vC8UyuJRC2n0h04
bP4UHoP41cwted2w5RcjdTmCguvKf5U/ov85UNhNgCk6WU/yXQSPGntkCqU
MmksyuzU6p53v/R0oXud+*C90maDRiBY+emXKn1HEpkRgJdEc4FznKebmEX7N
FO+2E+0KjUnogEDjancjC1G3bL80i1B4E1lgFFnHhUmoJm0ITGnFatr+poz
OHX97Cm0c1Fq9S60U0aXmGh74SquEMLQ6GehhNc+Uyq84HU166duE3LdghEKQ0
dLQZFN1cml1e9A0ZFN1cml1e9BHUeasS2U5K8A8ZFN1cml1e0BjVS53b20+ih0E
XcE1AC1FaktWb71CGwMGcwk1BwMGBH1AgkKcWqMBAh4BAheA00JE09ULxCT
A1UC81IA/0ZK1ks/ZcWUk36rj7rCgo/n+lgappSatum0htF1vGnAPoMbosBM3JG
1Fz0eJ5F+SGLN811D00hDRE3Y1dRy7Kc0RULm+92h0P2E0eS7G1M0R/cv
V+dEYFYCYZMYRppkUx6FAF9B78woF9eX8QRh62aB9h1QGup1kyUlnJMHdVbZ9U
vU1NKSBNChc7H6ZqU+h0yFATA/6V9zppqZ3gy/Dub79S2+QU20Fc3nr30kC1c
wBn1h61oEZ1s9Aqow0eB91ASXBzdZ0WUN9UB9pa7DJUmqPvEnjUmtUGGzd7F1fS2
haZ1Z1tRS0tp4Dv03JUuWdL8xfw0hK/c7H/9/Wd/c822FY02NYD7uRTwh1RCQ0
7Bm00RmusLL/9IM7G/s6jcc4bhsge/0v01j1gZPb6eRkRk070UhbzPa
5No1r0ADBQf8C8RP21qqdJ84DFkcU1z8p8yCJZRh86f0Qusefex0TmJn1x6s5S8
5u02Tnh6L3yZUP6431bt+hkaV2GHHeZFzVvJN8h1UaBKsYv07h6GH1Bg00vLX
JZV2uhIpg50jFeEydUJpg0ngD1L4uzJCT1Pk1y1mPg/M5PPr1UGS6HVFvME
H/T0mJd8FTYcU53Ja1Ce9U0v+VhmbwK6rQF9qngYc007+UQfNEwQHBeSacjMnd1
pml2DEK8eznnh+7P5Q4ARWk9A03S+ml9340ad888zU9CD00k1me1z13
0d93fGC88UDZ00sX8D10evn40B1Pux50TVhhBGRCAAJBQJLUn+0HsM0A0JE09U
LxCIAVtBCEA/0HGMKRNuBnLn1EpK6/10hkkrcfPHFnnbzyn/xj1qs9AP9o3je+
uW/13sU0P5nq1r5nbnMNPfQy4h53z2pna==
-----END PGP PUBLIC KEY BLOCK-----
```

The subsequent `--export` command in the example above is used to export the local *public* key named dSeries@ca.com from Computer A, allowing you to see the stored key data.

To send Computer A's public key to Computer B, simply re-run that `--export` command, but this time pipe the output to a file that can then be sent to Computer B...

```
C:\Program Files\GNU\GnuPG>gpg --armor --export dSeries@ca.com > dSeries.pk
```

There is a `--list-keys` command that enables you to list all the keys that are currently installed in the computer's local key store...

```
C:\Program Files\GNU\GnuPG>gpg --list-keys
C:\Documents and Settings\Administrator\Application Data\gnupg\pubring.gpg
-----
pub   2048D/930355AD  2010-01-20
uid   dSeries (dSeries GPG Key) (dSeries@ca.com)
sub   2048g/D4AE9161  2010-01-20

pub   1024D/C9C1D837  2010-01-20
uid   Alan Monument (Alan's CA Work e-Mail) (alan.monument@ca.com)
sub   1024g/27765DB7  2010-01-20
-----
```

Now that you have a local GPG *public* key, plus the imported *public* key of the person/organisation that you want to encrypt files for; you will now have to *bind* these together, as in the following steps...

```
C:\Program Files\GNU\GnuPG>gpg --edit-key alan.monument@ca.com
gpg (GnuPG) 1.4.10; Copyright (C) 2009 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub 1024D/C9C1D837  created: 2010-01-20  expires: never      usage: SC
                                trust: unknown  validity: unknown
sub 1024g/27765DB7  created: 2010-01-20  expires: never      usage: E
[ unknown ] (1). Alan Monument <Alan's CA Work e-Mail> <alan.monument@ca.com>

Command> fpr
pub 1024D/C9C1D837 2010-01-20 Alan Monument <Alan's CA Work e-Mail> <alan.monument@ca.com>
Primary key fingerprint: 6A30 0465 2E59 C9DF 976B 80DD 046F 75B7 C9C1 D837

Command> sign

pub 1024D/C9C1D837  created: 2010-01-20  expires: never      usage: SC
                                trust: unknown  validity: unknown
Primary key fingerprint: 6A30 0465 2E59 C9DF 976B 80DD 046F 75B7 C9C1 D837
      Alan Monument <Alan's CA Work e-Mail> <alan.monument@ca.com>

Are you sure that you want to sign this key with your
key "dSeries <dSeries GPG Key> <dSeries@ca.com>" (930355AD)?

Really sign? (y/N) y

You need a passphrase to unlock the secret key for
user: "dSeries <dSeries GPG Key> <dSeries@ca.com>"
2048-bit DSA key, ID 930355AD, created 2010-01-20

Command> quit
Save changes? (y/N) y

C:\Program Files\GNU\GnuPG>gpg -u dSeries@ca.com -r "Alan Monument" -o c:\temp\Encrypted_Log -e "C:\temp\ESP_System_Agent_R7_Install_IA.log"
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 1 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: depth: 1 valid: 1 signed: 0 trust: 1-, 0q, 0n, 0m, 0f, 0u
```

Note that all of the above setup steps only need to be done just once during the initial setup.

The last command is the one you would run on demand on Computer A when you want to encrypt a file (in this case the file *ESP_System_Agent_R7_Install_IA.log*) using the local *dSeries* private key to create an encrypted file named *Encrypted_Log* (you can use any meaningful name) using their public key (in this case Alan Monument). The encrypted file can then be sent to that user for decrypting at their end.

In summary, you previously run GPG to create a key store on Computer B, and then again to create the local private and public key combinations for that computer, followed by exporting the new public key which was sent to Computer A and imported using the steps above.

Listing the keys on Computer B shows the current key store state (it currently contains just the original single key)...

```
C:\Program Files\GNU\GnuPG>gpg --list-keys
C:/Documents and Settings/mona103/Application Data/gnupg/pubring.gpg
-----
pub 1024D/C9C1D837 2010-01-20
uid                Alan Monument <Alan's CA Work e-Mail> <alan.monument@ca.com>
>
sub 1024g/27765DB7 2010-01-20
```

You now need to import Computer A's public key that was exported earlier so that this computer can encrypt files to send in the other direction...

```
C:\Program Files\GNU\GnuPG>gpg --import dSeries.pk
gpg: key 930355AD: public key "dSeries <dSeries GPG Key> <dSeries@ca.com>" imported
gpg: Total number processed: 1
gpg: imported: 1
```

Listing the key store contents again you will now see two public keys...

```
C:\Program Files\GNU\GnuPG>gpg --list-keys
C:/Documents and Settings/monal03/Application Data/gnupg/pubring.gpg
-----
pub   1024D/C9C1D837 2010-01-20
uid   Alan Monument <Alan's CA Work e-Mail> <alan.monument@ca.com>
>
sub   1024g/27765DB7 2010-01-20

pub   2048D/930355AD 2010-01-20
uid   dSeries <dSeries GPG Key> <dSeries@ca.com>
sub   2048g/D4AE9161 2010-01-20
```

Again these above steps are only performed once to establish the environment.

When Computer A sends Computer B an encrypted file, all that needs to be performed on Computer B is to run the following command to decrypt that file...

```
C:\Program Files\GNU\GnuPG>gpg -u alan.monument@ca.com --output C:\temp\final_file
--passphrase-file C:\temp\PassPhrase.txt --decrypt c:\temp\Encrypted_Log
Reading passphrase from file descriptor 3

You need a passphrase to unlock the secret key for
user: "Alan Monument <Alan's CA Work e-Mail> <alan.monument@ca.com>"
1024-bit ELG-E key, ID 27765DB7, created 2010-01-20 <main key ID C9C1D837>

gpg: encrypted with 1024-bit ELG-E key, ID 27765DB7, created 2010-01-20
"Alan Monument <Alan's CA Work e-Mail> <alan.monument@ca.com>"
```

Note that the passphrase that was used when you originally created your *private* key is now needed to decrypt the file – this passphrase can be stored in a file that you create (in this case called *PassPhrase.txt*) which is used as in this example. The command completes with no other user interactive input required and creates an unencrypted file called *final_file* which you can then read and process in subsequent process steps.

Once you have proven that the above GPG commands interactively work correctly, you can then place an appropriate GPG command into a CA WA workload object as follows...

The screenshot shows the configuration window for a workload object named "Decrypt Using GPG". The "Basic" tab is active. The configuration includes:

- Name:** Decrypt Using GPG
- Qualifier:** (empty)
- Agent name:** AGENT
- Use Application-level defaults
- Run frequency:** A table with one row: Action: Run, When: DAILY.
- Use Application-level defaults
- Command to run:** C:\Program Files\GNU\GnuPG\gpg.exe
- Arguments to pass:** -u alan.monument@ca.com --passphrase-file C:\temp\PassPhrase.txt --output c:\temp\Final_File --decrypt c:\temp\Encrypted_Log
- User ID:** (empty)

As there are heaps of GPG commands and options at your disposal to perform encryption/decryption, there may be other command combinations that better meet your business requirements. You are therefore advised to read all the available documentation or defer to those people who know the GPG solution to provide you with an optimum approach.

JavaScript is not just for web pages

Contributed by Bob Pyette

JavaScript Introduction

There is nothing like a good scripting language when it comes to a clean, easy-to-use, high-level toolkit for both the programmer and the non-programmer alike. Generally, scripting languages are easier and faster to code in than the more structured and compiled languages such as C++ and Java, yet they still lend themselves to a wide range of applications.

JavaScript is Netscape's scripting language often used in web-based development. It was originally designed to add interactivity to HTML pages you see via your web browser. Similar to IBM's REXX and other scripting languages, it provides support for variables, language constructs, such as "if" statements, "for" and "while" loops, and many other scripting elements. In 1997, the ECMA international standards body standardized the core portion of the language. The result was a language, technically called ECMAScript, that looks and feels like JavaScript without the browser-specific parts.

Rhino is an open-source implementation of JavaScript that is typically embedded into Java applications. Rhino is one component of the Mozilla project. Developers of Java-based applications, such as job scheduling systems and identity management systems, are now embedding Rhino JavaScript to offer scripting capabilities to their users.

Adding scripting to an application brings a number of advantages. The functionality of an application is easily extended using a common and powerful tool. Scripting languages provide a natural and concise method of specifying behaviour since they are programming languages, after all. Since scripts are text, they can be easily incorporated into XML documents.

So while you may not be a web developer, the chances of encountering JavaScript as part of your day-to-day job are increasing. JavaScript is not just for web pages.

This article will help you familiarize yourself with JavaScript. It introduces some basic concepts, describes how to create and run a simple JavaScript script, describes some common JavaScript elements, and provides some scripting examples on how to use these elements. Note that the term "JavaScript" is used throughout this article to refer to the JavaScript scripting language in general.

Java vs. JavaScript

Many users fear the name "JavaScript" because they think it is the same as Java. When you say they can use JavaScript to handle their requirements, often the reaction is "But we're not programmers". Java and JavaScript are two completely different languages. It is like comparing REXX to C++. Java, developed by Sun Microsystems, is a powerful and very complex programming language. JavaScript, on the other hand, is a powerful scripting language and can be thought of as a lightweight programming language.

JavaScript, unlike Java, is an interpreted language which means it does not need to be compiled. A JavaScript program, normally referred to as a JavaScript script, is lines of executable computer code.

Creating and Running a Simple Script

Since you may be writing and running JavaScript scripts from a variety of your own applications, this first example makes use of a general-purpose technique.

The simplest way to create and run a simple JavaScript script is to use a text editor on your PC, such as Notepad, code the script, save it with a .htm extension (not with a .txt extension), and then double-click or open the file to run it. Although JavaScript statements are not HTML, they can be included within an HTML document.

The example below shows a simple JavaScript script created using NotePad and stored as a .htm file.

```
<html>
<script language="JavaScript">
// This is my first JavaScript script
MyName ="Bob";
document.write("Hello " + MyName);
</script>
</html>
```

Basic JavaScript in an HTML Document

The actual code for the script is stored between the <script> and </script> tags. The <script> tag identifies the scripting language as JavaScript. The <html> and </html> tags indicate this is an HTML document. The script starts with a comment line indicated by //. In the script, a variable called MyName is assigned the value "Bob". A method called document.write outputs the string "Hello " followed by the value of MyName. When this script opens in your web browser, the result is the string "Hello Bob".

If you are using JavaScript within your own application, the application should generate the tags it needs. Some applications may even include their own JavaScript editors, allowing you to easily imbed built-in functions and other language elements, and test and debug your scripts prior to implementation.

Some General Notes

JavaScript is case-sensitive. You will need to get used to the fact that *NAME*, *Name*, and *name* are all different. Watch your capitalization when referring to variables, objects, and functions.

Out of habit, some users will end each JavaScript statement with a semicolon. In general, semicolons are optional. However, they are required if you put more than one statement on a single line.

JavaScript ignores spaces. You can add white space (i.e. blank lines, indents, spaces between operators, etc.) to your script to make it more readable.

Basic Elements of JavaScript

This section introduces some basic elements of JavaScript, and includes some examples of JavaScript code.

Comments

You can add comments to your JavaScript script. A single-line comment starts with // and ends at the end of the line. A comment that spans multiple lines starts with /* and ends with */. You can include any number of comment lines between the comment delimiters.

Data Types

The basic JavaScript data types are:

- Numbers, such as 293 or 3.14. JavaScript supports both integers and floating-point numbers.
- Boolean (or logical) values, which can be either true or false.

- Strings, such as "Hello world".
- null (or empty) value, represented by the keyword null.

JavaScript is a loosely typed language. This means you do not have to specify the data type of a variable when you define it, and data types are converted automatically as needed during script execution. For example, in Figure 2, the variable Answer is first assigned the numeric value of 36. Then an expression concatenates a string with this number. JavaScript automatically converts the number to a string.

```
Answer = 36;
Fact = "Number of years since man first landed on the moon is " + Answer;
```

Variables

You can use variables to store and manipulate values in a script. You give variables names by which you refer to them. Variable names must begin with a letter or the underscore character. Subsequent characters can also include the digits 0 – 9. Some examples of valid names are CaMeL, _total, jan31, and APPLstart.

You can define a variable by simply assigning a value to it or by using the keyword var. For example, age = 65 and var age = 65, each assign the number 65 to a variable called age.

There are many functions, or methods, you can use to work with variables. Some examples will follow later in this article.

Literals

You use literals to represent fixed values. JavaScript supports the following types of literals:

- Integers, such as 47 or -852.
- Floating-point literals, such as 1.2345, -5.2E3, or 2E-5.
- Boolean literals, either true or false.
- String literals, such as "coconut", 'One small step for man', or "123".

To use special characters in strings, such as a quotation mark ("), an ampersand (&), or a backslash (\), you must "escape" the special character by preceding the character with a backslash. For example, to assign the file path "c:\temp" to a variable called Mydir, use the following:

```
Mydir="C:\\temp"
```

Expressions

An expression is a combination of literals, variables, and operators that evaluates to a single value. An arithmetic expression evaluates to a number; a string expression evaluates to a character string; a logical expression evaluates to either true or false.

Operators

JavaScript supports many different types of operators you can use in your expressions. Some of the more common operators are listed below.

The comparison operators are: == (is equal to), != (is not equal to), < (less than), <= (less than or equal to), > (greater than), >= (greater than or equal to).

Remember to use = when assigning a value to a variable (e.g. MyName = "Bob"), and == when comparing values (e.g. if MyName == "Bob" ...). Confusing these two is a common mistake in JavaScript programming.

You can use the following logical operators: || (or), && (and), ! (not)

In addition to the standard arithmetic operators (+, -, *, /), you can use % for modulus (division remainder), ++ (increment by 1), and -- (decrement by 1). You can also use + as a string operator to concatenate strings and variables.

Conditional Statements

Very often when you write code, you need to perform different actions for different decisions. You can use the following conditional statements in your code to do this.

- if statement – includes a condition and an action to take if the condition is true.
- if...else statement – identifies the actions to take based on a condition being true and the condition being false.

If you want to use more than one statement after an if or else statement, you must enclose the statements in curly braces {}.

Figure 3 shows an example of conditional statements. If the variable Latejob has a value of "payroll" and the variable Hour is greater than 10, the Notify variable is assigned the value 'fredbloggs@mycompany.com' and the Severity variable is set to 2. Otherwise, Severity is set to 3.

```
if (LateJob == "payroll" && Hour > 10)
{
    Notify = 'fredbloggs@mycompany.com';
    Severity = 2;
}
else
    Severity = 3;
```

You can also use a conditional operator to assign a value to a variable based on some condition. Basically, it is a short alternative to coding a simple if...else statement where a variable is assigned different values. The syntax is:

```
variable = (condition) ? (true action) : (false action)
```

Figure 4 shows two approaches for assigning a variable different values based on a condition. If the remainder when MyNumber is divided by 2 is equal to 0, then assign the string "Even" to the variable named MyType. Otherwise, assign "Odd" to the variable named MyType.

```
//One approach
if (MyNumber%2==0) MyType = "Even";
else MyType = "Odd"

//Another approach
MyType = (MyNumber%2==0) ? "Even" : "Odd";
```

JavaScript also supports the switch statement, which allows a script to evaluate an expression and attempt to match the expression's value to a case label. This allows you to combine several tests of the same variable or expression into a single block of statements.

In Figure 5, the expression `DayOfWeekNumber` is evaluated once. The value of this variable is compared with the values for each "case" in the structure. If there is a match then the block of code associated with that case is executed. Use the "break" statement to prevent the code from running into the next case automatically. If there is no match to a case, the "default" code applies. For example, if `DayOfWeekNumber = 6`, control passes to "case 6" where the `Start_batch` variable is set to "2PM" and the `Late_end` variable is set to "9AM tomorrow".

```
switch (DayOfWeekNumber)
{
case 0:
  Start_batch = "10AM";
  Late_end = "5AM tomorrow";
  break;
case 6:
  Start_batch="2PM";
  Late_end = "9AM tomorrow";
  break;
default:
  Start_batch="4PM";
  Late_end = "6AM tomorrow";
}
```

Looping

To run the same block of code multiple times, you can use looping statements in your code. In JavaScript, there are the following looping statements:

- while – loop while a condition is true.
- do while – loop through a block of code once, and then repeat the loop while a condition is true. The condition is tested at the end of the loop rather than at the beginning.
- for – loop for a specified number of times. The "for" loop typically uses a variable (i.e. a counter) to keep track of how many times the loop has executed, and it stops when the counter reaches a certain number.

A simple example of a "for" loop is shown in Figure 6. The variable `pyramid` is initialized to 0. In the "for" statement, the initial expression is `i=1`, the condition is `i<10`, and the increment is `i++` (add 1 to `i`). A left brace is used to signal the beginning of a block; a right brace is used at the end of the block. All the statements between the braces are executed with each iteration of the loop. Each time through the loop, the value of the counter `i` increases by 1 and the value of `pyramid` increases by the value of `i`. The loop stops when the condition `i < 10` is false. The result is a display of a series of 9 numbers: 1 3 6 10 and so on, where the difference between two consecutive numbers increases by 1.

[Insert Figure 6 – Looping with the for Statement]

```
pyramid = 0;
for (i=1; i<10; i++)
{
  pyramid = pyramid + i;
  document.write(pyramid, " ");
}
```

```
}
```

Note that a shortcut you could use for incrementing the value of `pyramid` in this example is `pyramid+=i`, which takes the value of `pyramid`, adds `i` to it, and assigns the result back to `pyramid`.

Objects and methods

Objects allow you to combine several kinds of data (properties) and functions to act on the data (methods) into a single, convenient package. Some examples of built-in JavaScript objects are:

- Date object – used to work with dates and times.
- Math object – includes math constants and functions.
- String object - used to work with text.
- Array object - stores a set of values in a single variable name, where each value is an element of the array with an associated index number, ranging from 0 to the number of elements - 1. For example: `Month[0]`, `Month[1]`, ..., `Month[11]`.

Each property is basically a variable in itself, and is contained within the object. Each property can be assigned a value. You can use properties to store any type of data a variable can store. The general syntax for using a property is `object.property_name`. Methods are functions that are stored as properties of an object. The general syntax for using a method is `object.method_name()`.

In Figure 7, a variable called `Animal` is assigned the value "Hippopotamus". Subsequent statements use methods and properties to work with this variable, as follows:

- `HowLong` has the value 12 because `Animal.length` resolves to the length of the string "Hippopotamus".
- `AllCaps` has the value "HIPPOPOTAMUS". `Animal.toUpperCase()` converts the value of the variable `Animal` to all upper-case characters.
- `Shortform` has the value "Hippo". `Animal.substring(0,5)` resolves to a string containing the 1st – 5th characters of the variable `Animal`. Note that the first position in a string is position 0. `substring(start,end)` starts at position *start* and includes all characters up to but not including position *end*.

```
Animal = "Hippopotamus";  
HowLong = Animal.length;  
AllCaps = Animal.toUpperCase();  
Shortform = Animal.substring(0,5);
```

JavaScript Summary

This article introduced the JavaScript scripting language through descriptions and examples. Although I have only scratched the surface of this powerful language, you can use many of these basic scripting elements to create useful scripts.

There are many books and web sites available where you explore the JavaScript language in more detail. Detailed reference information on the ECMAScript language specification can be found at <http://www.ecma-international.org/publications/standards/Ecma-262.htm>.

JavaScript is a powerful scripting language that is finding its way into many applications outside of its traditional web-based applications. JavaScript is not just for web pages anymore.

Getting Support for CA WA DE

CA Support provides the technical expertise and self-service tools you need to effectively utilise your CA solutions. Our highly trained professionals deliver best practices and knowledgeable guidance in 16 languages from 12 global support centres. CA also offer proactive, preventive and customised support solutions focused on your specific business needs. With CA Support, you have easy access to trusted advice that is always available.

What you get

This topic discusses how to get registered and access the CA Support site (<http://support.ca.com>), and refers to CA WA DE specific support related resources such as a the knowledgebase, documentation, message boards, forums, etc. Here you will also discover your support entitlements and be given a step by step description of initially getting yourself registered to, and then accessing the CA Support resources via CA Support Online.

CA Business Critical Support for CA software consists of operational assistance and technical support and generally includes:

- Direct access to technical support and the ability to open and manage support incidents via CA Support Online or by telephone
- 24x7x365 support for Severity 1 incidents
- Access to CA Support Online (<http://support.ca.com>) for 24x7x365 online technical support and access to CA software product and documentation downloads, fixes, service packs, patch downloads, user groups, user forums, beta testing, FAQs, samples, webcast recordings and demos, usage tips. Technical updates and HYPER notification as such are made available by CA
- Interactive remote diagnostic support allowing technical support engineers to troubleshoot an incident securely through a real-time browser-based remote control feature.
- Upgrades for the CA software if and when CA makes them Generally Available. Any CA software so provided is subject to the same usage limitations and restrictions as the CA software originally licensed to the customer by CA.

A description of the CA Support service is available in the document *Working with CA Support* available at <http://support.ca.com>. This link also contains CA's **Support policies and terms**.

Severity 1 Cases

To ensure prompt attention of any "system down" situation affecting your production environment, you must call the Global Contact Centre to report a Severity 1 condition. After November 16th, 2013, the option to select "Priority 1" when you open or update a case on CA Support Online will be removed and replaced with instructions on how to contact CA by phone. This change reflects CA's long standing guidance to open Severity 1 issues with a phone call, helping the CA Support team to improve the speed and consistency of their response to you.

Telephone Support

CA Technical Support and Customer Services can be engaged by telephone or through CA Support Online at support.ca.com. Refer to the *Support Phone Directory by Country* document at <http://www.ca.com/us/support/phone.aspx> for telephone numbers and hours of operation.

Online Support

CA Support Online is available 24 hours a day to allow you to manage your support cases (i.e. log, view, set severity, update and close). CA Support Online also provides various other useful features, for example: a Knowledge Base search; product and documentation downloads; fixes, service pack and patch downloads; support utilities; automated notifications; access to user groups, user forums, beta programs; subscriptions for hyper notifications and CA Tech Insider newsletters, etc.

If you are not already registered, you will first need to get yourself registered before you can access the CA support site. It is highly recommended that you get yourself registered in advance of actually needing to avail yourself of the support systems in an urgent situation, as your registration request will go through an authorisation process that can take some time.

While working with CA Support Engineers on cases; email communications and a remote access capability can also be used to deliver support services. Secure File Transfer Protocol (FTP) and a secure FTP web client allow for the protected transfer of case file attachments to CA Support Online, with the web client additionally providing case file transfer protections from CA Support, and files are encrypted and stored securely while on CA systems. Optionally unsecure FTP transfers also are supported.

Getting Registered to CA Support Online

Using your preferred Internet Browser, type support.ca.com into the address field and press the enter key.

The following CA page should appear...

The screenshot shows the CA Support Online website. At the top right, the 'Register' link is circled in red. The page layout includes a header with the CA Technologies logo and navigation links. The main content area features a 'Support for Businesses & Partners' section with a carousel of images and text, a 'New Certifications Available' announcement, and a 'CA Support Online Important Notices' section. A sidebar on the right contains 'Acquired Products Support' and 'New Enterprise Customers' links. The footer includes links for about us, news, events, careers, rss feeds, legal, privacy policy, and sitemap, along with a copyright notice for 2012 CA.

CA Workload Automation DE (dSeries Edition) Cookbook

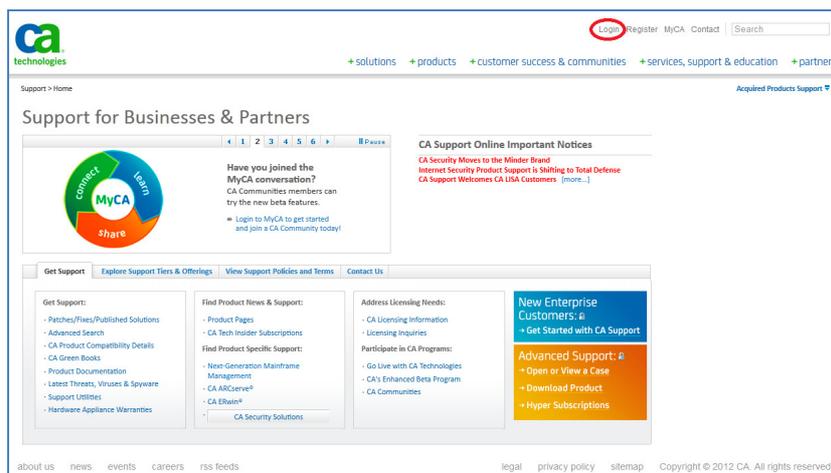
To start the registration process, simply click on the **register** link in the top of the pane (highlighted in the example above). You will then be presented with the **Register with CA Technologies** form.

Simply complete all fields with your details. Your e-mail address will need to be associated with the Company Name entered (i.e. not *name@msn.com*), and the Company Name needs to be associated with a valid six digit Customer ID.

Once submitted, the information provided on this form is validated and you will be granted access to the site. An e-mail will inform you when this process has been completed. You may now proceed to login to the CA Support site.

Logging in to CA Support

Using your preferred Internet Browser, type support.ca.com into the address field and press the enter key. This time click on the **login** link in the top of the pane (highlighted in the example below)...

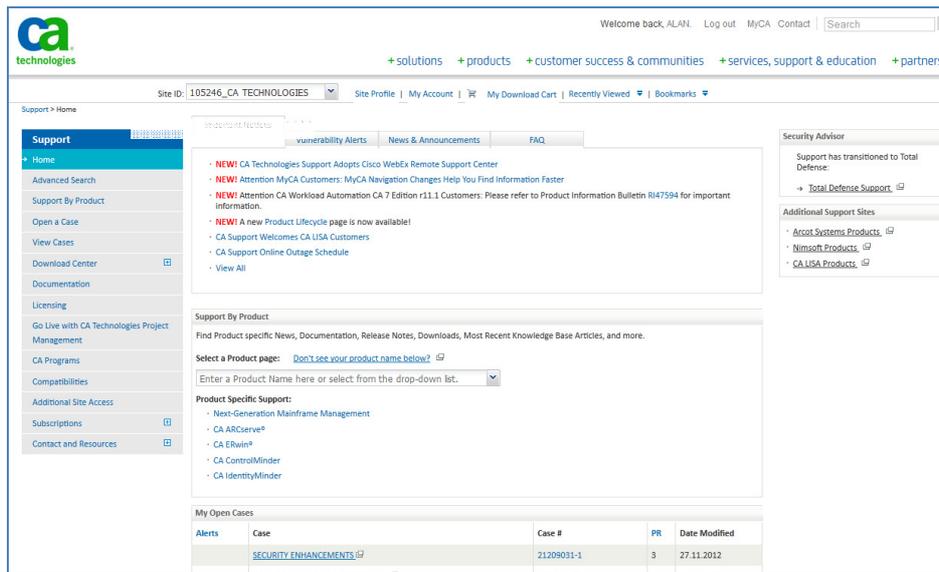


You will then be presented with the Login screen.

Enter the *Email Address* and *Password* details that you used when registering yourself, and click the **submit** button or press the Enter key.

CA Workload Automation DE (dSeries Edition) Cookbook

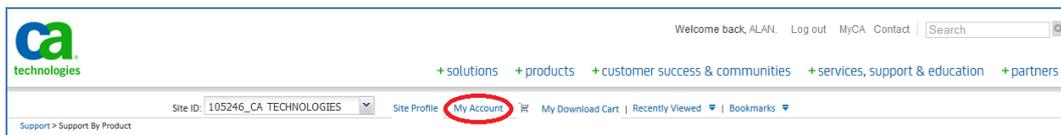
You will then be presented with the following **CA Support Online Home** page.



The left hand side of the page contains a navigation pane which you can use to quickly access the various parts of the Support site.

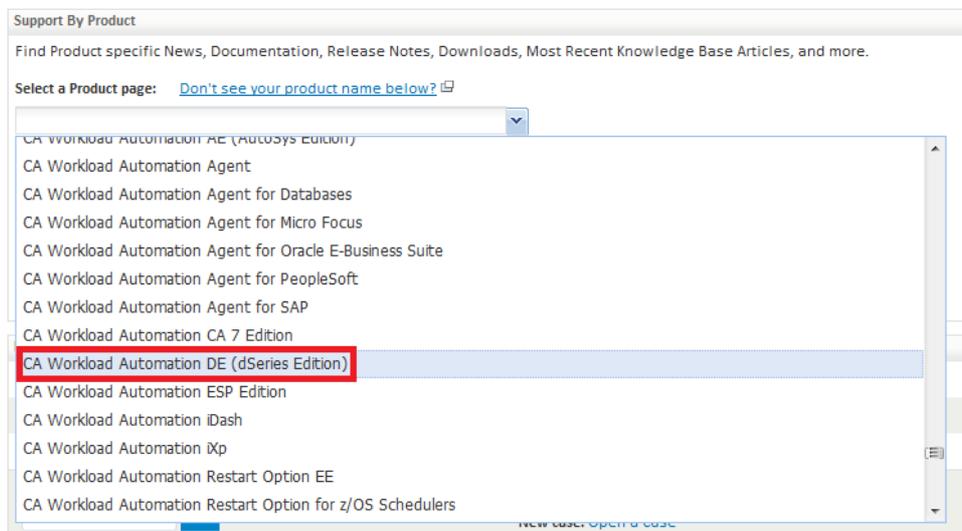
Updating Your Support Profile/Password

Should you need to change any of your profile details or change your password, simply click the **My Account** link at the top right of the *support.ca.com* page as highlighted below.



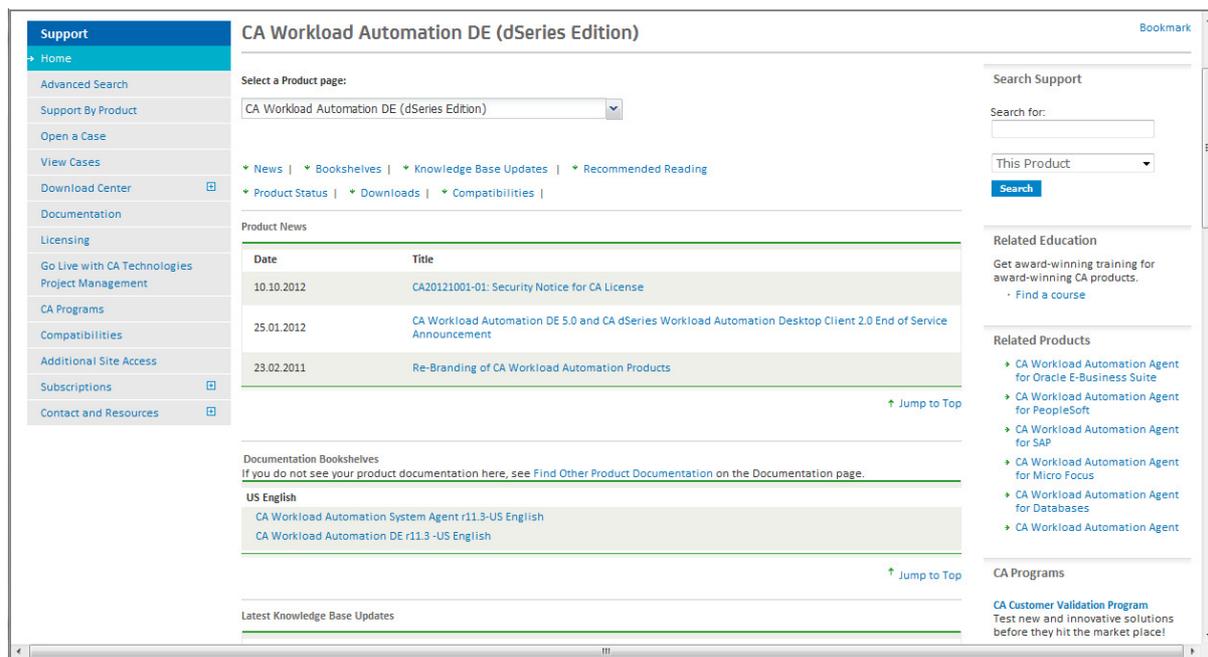
CA WA DE Product Home Page

To access a home page for a specific product, use the drop-down list in the *Support By Product* section in the middle of the window and click on the required product, **CA Workload Automation DE (dSeries Edition)**, as shown in the following example...



CA Workload Automation DE (dSeries Edition) Cookbook

The CA Workload Automation DE (dSeries Edition) product page will be displayed.



This page will usually be larger than your screen and you will need to use the scroll bar to see all the information. From this home page you will have access to:

- Support Knowledge Base – Use the Search Support pane in the top right
- Recent Product News
- Documentation Bookshelves to product guides, etc.
- Latest Knowledge Base Updates
- Recommended Reading
- Product Status – where you will find release and drop support announcements, product roadmaps, etc.
- Downloads – for product media and fixes
- Compatibility search to check that the solution works on your version and edition of operating system

Some of this information is presented in scrollable lists and displayed items are usually active links to the information. Simply move your mouse over any text in blue, and where the mouse pointer turns into a hand with a pointing finger you will be able to hyperlink to the actual information.

Clicking on any of the links in the *Documentation Bookshelves* section will present you with appropriate documentation which you can either view online or download to your workstation.

Further down the product home page you will find the *Downloads* pane which contains a drop-down list of only the CA Workload Automation DE product and options, as compared to requesting a Download from the *Download Center* off the navigation pane, where you may need to scroll through other CA solutions licensed to your organisation.

Navigation Pane

To the left of most CA Support Online pages you will find a navigation pane.

The navigation pane provides a fast way to navigate between the various services available on the CA Support site. Simply click on the desired area and you will be presented with an appropriate page.

The navigation pane also provides a visual notification of where you are on the site by highlighting the area you are in with a little arrow to the left of the current area, and also highlighting the area in a different colour.

Other pages you will find useful in addition to the Product Home Page are:

- [Advanced Search](#) – to search the support knowledge base
- [Open a Case](#) – to report a problem or ask a question to the CA Support team
- [View Cases](#) - to see all your open support cases
- [Download Center](#) – to download CA products and fixes
- [Documentation](#) – to view and download documentation
- [Licensing](#) – to report a licensing issue or request new keys

Support	
→ Home	
Advanced Search	
Support By Product	
Open a Case	
View Cases	
Download Center	⊕
Documentation	
Licensing	
Go Live with CA Technologies Project Management	
CA Programs	
Compatibilities	
Additional Site Access	
Subscriptions	⊕
Contact and Resources	⊕