# CA Workload Automation Agent for UNIX, Linux, Windows, or i5/OS

## Release Notes

r11.3 SP1, Cumulative 4

# CA Technologies Product References

This document references the following CA Technologies products:

- CA Workload Automation AE
- CA Workload Automation DE
- CA Workload Automation ESP Edition
- CA Workload Automation CA7 Edition
- CA Workload Control Center
- CA Workload Automation Desktop Client (CA WA Desktop Client)
- CA Workload Automation Agent for UNIX (CA WA Agent for UNIX)
- CA Workload Automation Agent for Linux (CA WA Agent for Linux)
- CA Workload Automation Agent for Windows (CA WA Agent for Windows)
- CA Workload Automation Agent for i5/OS (CA WA Agent for i5/OS)
- CA Workload Automation Agent for Application Services (CA WA Agent for Application Services)
- CA Workload Automation Agent for Web Services (CA WA Agent for Web Services)
- CA Workload Automation Agent for Micro Focus (CA WA Agent for Micro Focus)
- CA Workload Automation Agent for Databases (CA WA Agent for Databases)
- CA Workload Automation Agent for SAP (CA WA Agent for SAP)
- CA Workload Automation Agent for PeopleSoft (CA WA Agent for PeopleSoft)
- CA Workload Automation Agent for Oracle E-Business Suite (CA WA Agent for Oracle E-Business Suite)
- CA Workload Automation Agent for Remote Execution (CA WA Agent for Remote Execution)
- CA Process Automation

# Contact CA Technologies

**Contact CA Support**

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

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

**Providing Feedback About Product Documentation**

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

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

# Contents

## Chapter 6: Related Documentation 65

## Appendix A: Acknowledgements 95

# Chapter 1: Welcome

Welcome to CA Workload Automation Agent for UNIX, Linux, or Windows. This document includes information about changes to existing features, a complete list of fixed issues, supported platform information and system requirements, and known issues for this release, if any. Relevant documentation may also be included for significant fixes or changes.

For the latest version of the Release Notes, visit http://ca.com/support. Service pack releases do not feature a full updated documentation set with a bookshelf. You can find the Release Notes by searching the Find Other Product Documentation section on the Documentation page.

# Chapter 2: Changes to Existing Features

This chapter documents changes made to existing features in CA WA Agent for UNIX, Linux, Windows, or i5/OS r11.3 SP1.

This section contains the following topics:

## Encrypted Passwords Stored in Text Files

The agent's home directory and its subdirectories can contain text files with encrypted passwords. We recommend that you secure and restrict access to these directories and their contents.

## Native 64-Bit Support on Linux

The agent now has native 64-bit support on Linux. You can install a 64-bit agent on the following Linux platforms:

- Red Hat Enterprise Linux 4, 5 and 6
- SUSE Linux 10 and 11

## JRE Supplied with the Agent on AIX

The agent now supplies JRE 1.6 on AIX. If you are installing the agent on AIX, you are no longer required to install the JRE prior to the agent installation.

# Upgrade Option for Existing Customers

For existing customers, you can upgrade from r11.3 to r11.3 SP1 by applying the r11.3 SP1 roll-up, instead of performing a full installation. To apply the r11.3 SP1 roll-up, use the current patch utility. The patch installer updates the Java Runtime Environment (JRE) supplied with the agent and provides the same fixes and features as the full installer.

**Note:** Patch upgrade from Release 7 to r11.3 SP1 is not supported.

# Migration of Additional Release 7 Parameters

You can use the r11.3 SP1 agent installation program to migrate a Release 7 agent to r11.3 SP1. In the previous release, the installer did not migrate some common parameters from Release 7 to r11.3 during an agentparm.txt file conversion.

In r11.3 SP1, migration from Release 7 has been enhanced to include the following additional parameters:

- communication.timeout
- objmon.scaninterval
- core.health.monitor.enable
- core.health.monitor.interval
- filemonplugin.sleepperiod
- ftp.passive
- ftp.client.separator
- ftp.client.updatemsg
- ftp.data.compression
- log.archive
- log.level
- log.maxsize
- objmon.cpu.scalefactor
- runnerplugin.spool.clean.enable
- runnerplugin.spool.expire
- runnerplugin.spool.sleep
- persistence.coldstart

**Notes:**

- The following oscomponent parameters are *not* migrated from Release 7:
  - oscomponent.javapath
  - oscomponent.startjvm
  - oscomponent.attachjvm
  - oscomponent.classpath
  - oscomponent.libjvmpath
  - oscomponent.servicename
  - oscomponent.servicedisplayname

  - oscomponent.dumpenvironment
  - oscomponent.jvm
  - oscomponent.jvm.x.options
  - oscomponent.not.authenticate.su

  As a result, you can migrate from a 32-bit Release 7 agent to a 64-bit r11.3 SP1 agent on the same operating system.

- On i5/OS, the oscomponent.loginshell parameter is always set to true, regardless of the setting in the Release 7 agentparm.txt file.
- The log.allow.method parameter does not apply to the r11.3 agent and is not migrated.
- For more information about these parameters or migration, see the *Implementation Guide*.

## Separate Installer Properties Files on UNIX and Windows

Separate installer properties files are now provided on UNIX and Windows. On UNIX, the name of the file is unix_installer.properties. On Windows, the name of the file is win_installer.properties.

## Insertion of Additional Agent Parameters During a Silent Installation

When installing the agent using a silent installer, you can now specify additional agent parameters in a text file. The parameters are inserted at the end of the agentparm.txt file without any validation or modification.

To insert agent parameters at the end of the agentparm.txt file, include the following property in the installer properties file:

**RAW_DATA**

Specifies the path to and name of a text file that contains additional agent parameters. In the text file, list each parameter on a separate line. During a silent installation, these parameters are inserted at the end of the agentparm.txt file without any validation or modification.

### Example: Insert Parameters to Configure the Agent to Clear Spool Files Automatically

In this example, the installer properties file contains the following property:

```
RAW_DATA=/usr/home/joe/additional_agentparm.txt
```

The additional_agentparm.txt file contains the following agent parameters to configure the agent to clear spool files automatically:

```
runnerplugin.spool.clean.enable=true
runnerplugin.spool.expire=50000
runnerplugin.spool.sleep=20000
```

During a silent installation, these parameters are added at the end of the agentparm.txt file.

**Note:** For more information about installing the agent using a silent installer, see the *Implementation Guide*.

## Service Pack and Maintenance Level Information Added to Version Output

When you issue the cybAgent -v command, it now displays the service pack and maintenance level. You can use this information to determine the patch that is applied to the agent.

## Default Encryption Keys

In the previous release, the installation program did not provide default encryption keys. In this release, the installation program provides the following default encryption keys that are based on the cipher algorithm you choose:

- DES—0xC468284E7BF6FB41
- DESEDE—0xB6F98D9B6419D51A41FEEEE847E27DC01F2B1ABE058142E3
- AES—0x32A26EA95ACC64B1B19DAF713D60DC10
- BLOWFISH—0xC832DD138ECCDC8A70F88E3120C6AA2F0148D725B19B3DF670B23E8514D6EC6B

**Note:** These keys were generated at random and have no set pattern.

## Removal of Windows Short Cuts

The Windows short cuts have been removed to resolve an issue with short cuts being overwritten in a multiple agent installation.

**Notes:**

- To start or stop the agent, use the Windows Service Control Manager.
- To install the agent service manually, use cybagent -install at the command prompt.
- To remove the agent service, use cybagent -remove at the command prompt.
- To uninstall the agent, use Add/Remove Programs (Uninstall a program), similar to other Windows applications.

## Uninstall Mode

In the previous release, if you install the agent using the silent installer, the uninstall is performed in silent mode without user notification. In this release, this behavior has changed. If you uninstall the agent, by default GUI mode is now used on Windows and console mode is now used on UNIX.

## MD5 Sum Information Added to Patch Install Log

The patch installer log (patch_install.txt) now includes the MD5 sum of each file that is installed. This information is also stored in the MD5SUM.txt file, which is located in the agent installation directory. You can use this information to determine which files have changed since the previous patch was applied.

# Enhanced Job Cancellation on UNIX

By default, when you cancel a UNIX job, the agent sends the kill signal to the script's process group ID. As a result, the kill signal is delivered individually to all processes that are members of the group. Any process that is not a member of the group does not get terminated.

To ensure that all processes associated with a UNIX job are terminated, set the following parameter in the agentparm.txt file:

```
oscomponent.terminate.subtree=true
```

Instead of killing the process group, the agent will assemble the process subtree and manually kill each process.

**Note:** If oscomponent.terminate.subtree is set to true, the agent sends the SIGKILL signal to cause the process to terminate immediately. The agent does not provide the option to send the SIGTERM signal, so the receiving process cannot perform any clean-up upon receiving the signal. The default behavior (oscomponent.terminate.subtree=false) does provide an option to send the SIGTERM signal.

# Chapter 3: Known Issues

The chapter details the known issues in CA WA Agent for UNIX, Linux, Windows, or i5/OS r11.3 SP1.

This section contains the following topics:

## Patch Installer Fails on AIX

**Valid on AIX**

**Symptom:**

When I run the patch installer on AIX, it fails. The patch installer log file contains an error message similar to the following:

```
An IOException occured.
java.io.FileNotFoundException:
/home/yyyy/R113/AGENT/libfilefilter.so (Text file busy)
```

**Solution:**

The patch installer can fail if an agent shared library is still active in the system memory.

**To correct this problem**

1. Stop the agent:

   ```
   ./cybAgent -s
   ```

2. Issue the following command as root to remove any unused shared objects from memory:

   ```
   slibclean
   ```

3. Rerun the patch installer.

# CA WA Agent for Remote Execution Stops Working after Upgrade

**Symptom:**

After I upgrade the agent to r11.3 SP2 using the patch utility, CA WA Agent for Remote Execution stops working.

**Solution:**

During the patch upgrade, the third-party library sinetfactory.jar is replaced by an older version that is included with the agent. To correct this problem, restore the sinetfactory.jar file with the version that CA WA Agent for Remote Execution requires.

**To correct this problem**

1. Stop the agent:

   - On UNIX:

     *agent_install_path*/cybAgent -s

   - On Windows:

     *agent_install_path*\cybAgent -s

   ***agent_install_path***

   Specifies the agent installation directory.

2. Copy the sinetfactory.jar file from the backup directory created by the patch utility:

   - On UNIX:

     *agent_install_path*/patches/*patch_id*/backup/jars

   - On Windows:

     *agent_install_path*\patches\*patch_id*\backup\jars

   ***patch_id***

   Specifies the ID of the patch that is being applied. The ID is usually the name of the file that is downloaded from CA Support Online.

   You will use this file to replace the sinetfactory.jar file that is included with the agent.

3. Locate the sinetfactory.jar file in the agent installation directory:

   - On UNIX:

     *agent_install_path*/jars

   - On Windows:

     *agent_install_path*\jars

4. Replace the sinetfactory.jar file in this directory with the version that you copied in Step 2.
5. Restart the agent:
   - On UNIX:

     ```
     agent_install_path/cybAgent &
     ```

   - On Windows:

     ```
     agent_install_path\cybAgent -a
     ```

   The sinetfactory.jar file is restored.

# Unable to Start Agent after Configuring the Agent for WebLogic

**Valid on AIX and z/Linux**

**Symptom:**

When I configure the CA WA Agent for Application Services plug-in for WebLogic, my agent fails to start. The issue occurs when the SNMP connector is enabled on the agent.

**Solution:**

If you configure the plug-in for WebLogic using the instructions in the *CA Workload Automation Agent for Application Services Implementation Guide*, the following agent parameters will be set:

```
javax.xml.transform.TransformerFactory=com.sun.org.apache.xalan.internal.xsltc.tr
ax.TransformerFactoryImpl
javax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SAXPar
serFactoryImpl
javax.xml.parsers.DocumentBuilderFactory=com.sun.org.apache.xerces.internal.jaxp.
DocumentBuilderFactoryImpl
```

On agents using IBM JVMs (AIX and z/Linux), these values are incorrect.

**Note:** The preceding values are correct for agents using non-IBM JVMs (HP-UX, HP-Itanium, Solaris, and Linux).

**To correct this problem on AIX and z/Linux**

1. Configure the following parameters in the agentparm.txt file to the values shown:

   ```
   javax.xml.transform.TransformerFactory=com.ibm.xtq.xslt.jaxp.compiler.Transfo
   rmerFactoryImpl
   javax.xml.parsers.SAXParserFactory=com.ibm.xml.xlxp.api.jaxp.impl.SAXParserFa
   ctoryImpl
   javax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBuild
   erFactoryImpl
   ```

2. Start the agent.

# i5/OS Job Fails with Spool File Reading Failure

**Valid on i5/OS**

When running an i5/OS job, the job sometimes fails with the following error message:

```
...MAIN STATE FAILED SetEnd Cmpc(4001) Status(Spool file reading failure. Unable to
retrieve *USER return code. See previous messages in the log.) LStatus(Spool file
reading failure. Unable to retrieve *USER return code. See previous messages in the
log.)...
```

In this issue, the defaultlog_agent.log file located in the agent installation directory contains the following error:

```
Exception reading joblog spool file:
com.ibm.as400.access.ClientAccessDataStream incompatible with
com.ibm.as400.access.NPDataStream
```

This issue is due to a known issue with a third-party library that the agent uses. This issue will be fixed in a future release.

# Text Monitoring Job Fails on i5/OS

**Valid on i5/OS**

When running a Text Monitoring job on V7R1, the job sometimes fails with the following error even though the file member was created successfully:

```
CPF4102 File … in library … with member … not found
```

This issue is due to a known issue with a third-party library that the agent uses. This issue will be fixed in a future release.

# chkusr Utility Not Supported on i5/OS

**Valid on i5/OS**

The chkusr utility that is provided with the agent is not supported on i5/OS because the standard PAM (Pluggable Authentication Modules) library that chkusr uses is not applicable to the i5/OS environment. This issue will be addressed in a future release.

# Chapter 4: Fixed Issues

The chapter details the issues that have been fixed in CA WA Agent for UNIX, Linux, Windows, or i5/OS r11.3 SP1.

## Command Job Fails When Using a Glob for Standard Input (Windows only)

**Valid on CA Workload Automation AE**

When you specify the standard input as a global binary large object (glob), the command job fails on Windows. This issue has been fixed.

## Agent Hangs During Shutdown (UNIX only)

To initiate and track the shutdown, the agent uses PIDs and the ps command, which are prone to error. In rare cases, the shutdown can hang due to the ps command not completing. This issue has been fixed.

## Delays in Job Submission Due to Retrieving Large Spool File

**Valid on Linux and Solaris**

When retrieving a large spool file, the submission of jobs can be delayed, causing the agent to appear to be unresponsive. Jobs are submitted when the spool file processing completes.

To resolve this issue, the following modifications have been made to the agent:

- The thread pool is used for spool file retrieval
- More tracing information is written to the plugin_log_runner.log file when log.level=8 to aid debugging
- Optimization of spool file retrieval to take fewer I/O requests, allowing the agent to complete retrieval requests faster

## Cannot Download Files from a Mainframe Server Using USS

When trying to download files from a mainframe server using UNIX System Services (USS) for z/OS, the jobs fail with the following error:

```
The system cannot find the path specified
```

In this issue, the agent incorrectly treated PDF files as mainframe data sets instead of as UNIX files. This issue has been fixed.

## Incorrect Output Produced in Windows When Using a UNIX-like Emulator (UNIX only)

When a UNIX-like emulator such as MKS Toolkit is used to execute a shell script, the job produces incorrect output in Windows. In this issue, the shell script performed operations using case-sensitive environment variable names, which the agent incorrectly converted to upper case. This issue has been fixed.

## Cannot Comment Out Variable Assignment in Profile File (UNIX only)

**Valid on CA Workload Automation AE**

The agent fails to start if a line in a profile file contains a commented out variable assignment, for example:

```
#BL_USER_SPECIFIC_LAB001_VAL_2=user_specific_LAB001_val2
```

This issue has been fixed.

## Cannot Retrieve Standard Output File When Initial Working Directory is Set to User (UNIX only)

When the initial working directory is set to USER (oscomponent.initialworkingdirectory=USER) in the agentparm.txt file, the agent cannot locate the standard output file. This issue has been fixed.

# Excessive World Writeable Privileges (UNIX only)

The following files that the agent creates on UNIX have excessive world writeable privileges, which can conflict with auditing policies:

- Temporary working shell scripts (CA Workload Automation AE only)
- Standard output and standard error files
- Job logs
- Spool files

To customize the permissions of these files, we highly recommend that you add both of the following parameters to the agentparm.txt file:

**oscomponent.defaultfile.permission**

Specifies the standard UNIX file permission in octal notation starting with 0. The four-digit octal code specifies the default file access permissions for the following files that the agent creates:

- Temporary working shell scripts (CA Workload Automation AE only)
- Standard output and standard error files
- Job logs
- Spool files

**Example:** 0600 (grants read and write permissions to the owner, but prevents anybody else from accessing the file)

**Notes:**

- If oscomponent.defaultfile.permission is not specified, all files the agent creates will have the same permissions as before 11.3 SP1 cumulative 4.
- Temporary working shell scripts are granted execute permissions by the agent regardless of this parameter.
- This parameter does not change the access permission of the spool directory (that is, drwxrwxrwt).

**oscomponent.umask**

Provides support for the umask command, which turns off (disables) specific permissions that the oscomponent.default.permission parameter allows. The three-digit octal code sets the file mode creation mask (umask) for the following files that the agent creates:

- Temporary working shell scripts (CA Workload Automation AE only)
- Standard output and standard error files
- Job logs
- Spool files

**Example:** 066 (assuming the default file access permission is 666, this value turns off read and write permissions for the group and others)

**Notes:**

- If oscomponent.umask is not specified, the default umask of the user that started the agent is used for job logs, spool files, and wrapping scripts.
- For standard output and error files, the default umask of the user that runs the job is used with an exception on AIX and HP-UX. On AIX and HP-UX, the default umask is only used if the umask is set in the user profile.

**Notes:**

- The oscomponent.defaultfile.permission parameter defines the baseline for file permissions. The umask value further restricts which permissions are allowed to determine the final file permission. The umask value can be set in the oscomponent.umask parameter, the user profile, the job profile, and other sources.
- For job logs, spool files, and wrapping scripts, the agent determines the final file permission using the oscomponent.defaultfile.permission and oscomponent.umask parameters.
- For standard output and error files, the agent determines the final file permission using the oscomponent.defaultfile.permission parameter and the umask value that takes precedence. For example, if you set the umask value in the user profile and job profile, the umask value in the job profile takes precedence. If the umask value is set to 022 in the user profile and 021 in the job profile, the final umask value is 021.
- If you redirect the output of the command in an argument, these parameters do not apply and the file permission depends on the operating system. For example, if you specify the command as "/usr/bin/echo" and the argument as "TEST >> /tmp/TEST.OUTPUT.COMMAND", the file permission of TEST.OUTPUT.COMMAND is unspecified.
- On CA Workload Automation AE, if you get a 4030 completion code, it means that the agent could not read or write to the temporary wrapper script the agent creates. To resolve the error, verify that the combination of oscomponent.defaultfile.permission and oscomponent.umask parameters give the owner at least read and write permission.

### Example: Customize the Permissions of the Agent Working Files on CA Workload Automation ESP Edition

In this example, the following agent parameters are set:

```
oscomponent.umask=113
oscomponent.defaultfile.permission=0664
```

When the agent creates the following files, the permissions are set as indicated in parentheses:

- Job logs (-rw-rw-r--)
- Spool files (-rw-rw-r--)

If no user is specified in the job, the permission of the standard output and error files is -rw-rw-r--. If a user is specified in the job with a default umask of 022, the permission of the standard output and error files is -rw-r--r--.

### Example: Customize the Permissions of the Agent Working Files on CA Workload Automation AE

In this example, the following agent parameters are set:

```
oscomponent.umask=066
oscomponent.defaultfile.permission=0600
oscomponent.noforceprofile=true
oscomponent.cmdprefix.force=true
oscomponent.profiles.src.delay=true
oscomponent.profiles.global.override=true
```

The job profile has a umask value of 111.

When the agent creates the following files, the permissions are set as indicated in parentheses:

- Temporary working shell scripts (-rwx--x--x)
- Standard output and standard error files (-rw-------)
- Job logs (-rw-------)
- Spool files (-rw-------)

# File Watcher Jobs Fail if Watch File Path Contains Spaces or UNC Format

**Valid on CA Workload Automation AE**

When running a File Watcher job, the job fails if the path specified in the watch_file attribute contains spaces, for example:

```
insert_job: fw11
job_type: FW
machine: winagent
watch_file:"C:/Program Files/CA/WA Agent R11.3.2/filewatcher.exe"
owner:Administrator@winagent
```

In addition, the job fails if the path specified in the watch_file attribute is a UNC path, for example:

```
watch_file: \\CYBNT\MyDesktop\notify.txt
```

These issues have been fixed.

# User Verify Command Fails (Windows only)

If the following parameters are set in the agentparm.txt file, the User verify command fails on CA Workload Automation AE:

```
oscomponent.cmdprefix.force=true
oscomponent.cmdprefix.force.quotes.full=true
oscomponent.lookupcommand=true
```

In addition, Windows jobs fail on other scheduling managers with these settings if no user is specified in the job definition.

This issue has been fixed.

## Text Monitoring Job Succeeds When the Search String Occurs Outside the Date Range

A Text Monitoring job with date/time search mode succeeds when the line containing the searched string is beyond the upper boundary of the date range. This issue has been fixed.

In addition, a new parameter was added for text searches based on line numbers:

**objmon.textmon.lines.upper.include**

Indicates whether to include the upper line number (TO) in the search range for Text Monitoring jobs.

**true**

Includes the upper line number (TO) in the search range (inclusive).

**false**

Excludes the upper line number (TO) from the search range (exclusive).

**Default:** false

**Note:** This parameter does not affect text searches that are based on date/time or regular expressions, which always include the upper boundary in the search.

## Partial Job Output When Using Same File for Standard Output and Error (UNIX only)

If you specify the same file for standard output and standard error, the job output is partial. The issue occurs when you use > to overwrite the contents of the standard output and standard error files, for example:

```
std_out_file: > /tmp/avt_c_test.log
std_err_file: > /tmp/avt_c_test.log
```

After the fix, the information in the standard output and standard error files is complete.

# Refused by Agent Security When Running File Watcher Jobs

**Valid on CA Workload Automation AE**

When you run a File Watcher job with local security enabled (security.level=on), the job is refused by agent security. This issue does not occur with File Trigger jobs.

You can now set up your security.txt file to allow File Watcher jobs to run with local security enabled.

### Example: Allowing File Watcher Jobs to Run on UNIX

To allow File Watcher jobs to run on UNIX, allow access to the filewatcher executable and the directory where the spool and log files are created.

The following security.txt file allows File Watcher and UNIX command jobs to run:

```
c a * * *
f a * * +
x d * * +
x a * * /CA/WA_Agent_R11_3/filewatcher
x a * * /CA/WA_Agent_R11_3/spool/SCH/MAIN/WAAE_WF0.1/*
```

**Note:** When running File Watcher and UNIX command jobs, the agent creates temporary shell scripts in the spool directory. As a result, if you allow File Watcher jobs to run, you cannot prevent UNIX command jobs from running.

### Example: Allowing File Watcher Jobs to Run on Windows

To allow File Watcher jobs to run on Windows, allow access to the filewatcher.exe executable.

The following security.txt file allows File Watcher jobs to run, but prevents Windows command jobs from running:

```
c a * * *
f a * * +
x d * * +
x a Admin* Admin+ C:\Program Files\CA\WA Agent R11.3\filewatcher.exe
```

# Support for Running Windows Interactive Jobs in Session 0

In Windows XP/Server 2003 and prior versions, both services and applications used the same session (Session 0). As a result, Windows services could run interactive applications in the same session as user applications. Users could also run applications in other sessions using Terminal Services (Remote Desktop).

Windows applications (GUI and console) run in an environment called Windows Station/Desktop. Window Station/Desktop provides the resources and environment (such as fixed heaps) necessary for the applications to run. The resources provided by each Window Station/Desktop is limited. Some applications (such as Microsoft Excel) require more resources than non-interactive Window Station/Desktop can provide. In that case, the job needs to be run interactively.

When running interactive jobs, the user logs in to the interactive Windows Station/Desktop or connects to it using the Remote Desktop. The interactive Windows Station/Desktop receives user mouse/keyboard input and is visible to the user.

To run interactive jobs, you can use Terminal Services or Session 0.

## Terminal Services

Starting with Windows Vista/Server 2008, services run in Session 0 and applications run in session 1 and above (Terminal Services).

By default, the agent runs interactive jobs using Terminal Services. To run interactive jobs using Terminal Services, the following conditions must be met during job submission:

- The user that is specified to run the job must be logged in to the agent computer, either using Remote Desktop or locally
- The user must be logged in to the agent computer exactly once

If these conditions are not met, the job fails with one of the following submission errors:

```
Unable to locate interactive logon session
```

```
Multiple active logon sessions
```

## Session 0

To overcome these limitations, the agent now supports running Windows interactive jobs in Session 0. To run Windows interactive jobs in Session 0, set the following parameter in the agentparm.txt file:

```
oscomponent.interactive.sessionzero=true
```

With this setting, an interactive job runs in Session 0. In Windows XP/Server 2003, the application window displays in the same way as user applications. In Windows Vista/Server 2008 and newer versions, the Interactive Services Detection service must be running. A dialog displays asking the user to switch to that application. Only a single pair of Windows Station/Desktop is available.

Running interactive jobs in Session 0 have the following disadvantages:

- Application has to be granted access to run in that session. Thus, you can potentially run into issues with the limited number of entries in the Window Station/Desktop Access Control Lists (ACL).
- In Windows XP/Server 2003, if the user logs on/off interactively, the ACL can be reset. If the user application tries to create more windows (require more access to the Window Station/Desktop), the application can fail.
- If the agent is restarted, it loses track of the logon sessions. Thus, the agent is unable to clean up adequately after applications complete their execution.
- Since there is only one Window Station/Desktop, applications can run out of User/GDI heap, thus limiting the number of concurrently running jobs. We recommend that you limit the number of concurrent running jobs to no more than 10 at a time.

**Important!** Microsoft has determined that using interactive programs in Session 0 is a security vulnerability.

http://msdn.microsoft.com/en-us/windows/hardware/gg463353.aspx

Starting with Windows Server 2008 R2, the Interactive Services Detection service is not started automatically. You must manually start the Interactive Services Detection service.

Starting with Windows Server 2012, the Interactive Services Detection service cannot be started without a change to the system registry. Microsoft warns that making this registry change has the potential to destabilize the operating system. This option may not be supported in the next release.

As long as Microsoft continues to provide backward compatibility, CA will continue to support this functionality even at the expense of system security and stability. CA urges all Workload Automation customers find alternatives to using Session 0 for their Workload Automation tasks.

## Additional Parameters

Depending on your environment, you may also require the following parameters to use Session 0.

**oscomponent.su.newconsole**

(Optional) Sets whether to force the agent to create a console when running a Windows job.

**true**

Creates a console when running a Windows job.

**false**

Does not create a console when running a Windows job.

**Default:** false

**oscomponent.cmdprefix.force.redir.inline**

(Optional) Sets whether the agent forces inline redirection. In some cases, such as Windows Server 2008 R2, inline redirection is required for the console to be displayed in Session 0.

**true**

Forces inline redirection using cmd.exe /c *command args stdin_file* >*stdout_file* >*stderr_file*.

**false**

Does not force inline redirection.

**Default:** false

**Note:** This parameter requires oscomponent.cmdprefix.force=true and affects FTP verbose mode. If regular redirection is used, the command ftp –s:command_file will not output raw FTP commands with completion codes as per FTP protocol.

**oscomponent.cmdprefix.force.redir.ifstdon**

(Optional) Sets whether to prevent the agent from redirecting STDOUT/STDERR if none are specified in the job definition.

**true**

Prevents the agent from redirecting STDOUT/STDERR if none are specified in the job definition. If the output of the command requires user input, such as pause, it is displayed in the cmd.exe console window.

**false**

Does not prevent the agent from redirecting STDOUT/STDERR if none are specified in the job definition.

**Default:** false

**Note:** This parameter requires oscomponent.cmdprefix.force=true and the Windows job has to be interactive.

**Note:** If you are not logged in when the job is submitted, you might need to restart the "Interactive Services Detection" service to receive the prompt.

**More information:**

Configure the Agent for Windows Interactive Jobs (see page 77)

## File Watcher Job Fails When the Watch File is a Variable

**Valid on CA Workload Automation AE**

If a variable is specified in the watch_file attribute for a File Watcher job, the job fails with the following error:

```
Invalid format, Name should represent full path
```

The issue occurs when the name of the file does not contain the full path and does not apply to File Trigger jobs. This issue has been fixed.

## Job Output Gets Truncated When Argument Ends with a Backslash (Windows only)

When a command job is run under a user with a slash (\) at the end of the argument that is passed to the command, the job output gets truncated. This issue has been fixed.

## Jobs Fail When Writing to STDOUT or STDERR Files Larger than 2 GB (UNIX only)

When a 32-bit agent attempts to write to a standard out or standard error file with a size greater than 2 GB, the job fails with a "File too large" error. This issue has been fixed.

## Job Output Does Not Include All the FTP Generated Output (Windows only)

If you run a job that performs an FTP, the output of the job does not include all the FTP generated output. As a result, the output of the job does not match the output of FTP when executed at the command line.

You can now add the following parameters in the agentparm.txt file to include all the FTP generated output:

```
oscomponent.cmdprefix.force.redir.inline=true
oscomponent.cmdprefix.force=true
oscomponent.lookupcommand=true
```

## Agent Fails to Start After Applying Windows Security Patches

After applying Windows security patches to the agent computer, the agent fails to start. This issue occurs when the agent crashes, causing files in the database directory on the agent to get corrupted. This issue has been fixed.

## Agent Fails to Start After Rebooting Server (UNIX only)

When rebooting the server that the agent is running on, the agent sometimes fails to start. This issue occurs when the PID that is specified in the agent status file gets used by some other process. After the fix, the agent starts if it is not already running regardless of the PID in the status file.

# Failure to Process KILLJOB Events

**Valid on CA Workload Automation AE**

When using the term_run_time attribute to terminate a job automatically after a specified number of minutes, an error similar to the following message can occur:

```
CAUAJM_I_40245 EVENT: CHK_TERM_RUNTIME  JOB: test_sleepterm
CAUAJM_W_40032 Another KILLJOB for job test_sleepterm is in progress. Nothing to do.
```

This issue occurs when the max_run_time KILLJOB event is processed after the job completes, but before CA Workload Automation AE has been updated that the job is complete. A KILLJOB event that is issued in this time window between the actual completion and the CA Workload Automation AE update is not correctly processed. The unprocessed KILLJOB event prevents any additional KILLJOB events from being issued for that job. After you apply the associated fix on CA Workload Automation AE, this timing issue can no longer occur.

# Cannot Override the Initial Working Directory Using the PWD Environment Variable

If the oscomponent.initialworkingdirectory agent parameter is set, the initial working directory cannot be overridden in the job using the PWD environment variable. After the fix, the PWD value that is set in the job definition overrides the oscomponent.initialworkdingdirectory setting on the agent, for example:

In the agentparm.txt file:

```
oscomponent.initialworkingdirectory=USER
```

In the JIL:

```
command: dir
envvars: PWD=C\:\temp
```

Before the fix, the output from the dir command is from the user's home directory. After the fix, the output from the dir command is from the C:\temp directory.

# File Watcher Job Fails When Directory Does Not Exist (Windows only)

**Valid on CA Workload Automation AE**

If the directory of the monitored file in a File Watcher job does not exist, the job fails before running with the message <Scan Failed>. No job log files are generated on the agent. After the fix, the File Watcher job continues to run if the specified directory does not exist.

# Enable AES 256-bit Encryption for CA Workload Automation CA7 Edition

The agent now supports AES 256-bit encryption for messages that are sent between the agent and CA Workload Automation CA7 Edition.

**Note:** CA Workload Automation CA7 Edition requires IBM z/OS Integrated Cryptographic Service Facility (ICSF) on the mainframe. To set up requirements on CA Workload Automation CA7 Edition, see the *CA Workload Automation CA7 Edition AES256 Enhancement* document.

By default, the Java Cryptography Extensions (JCE) permits the use of 128-bit keys with the strong Jurisdiction Policy Files. Thus, the agent permits the coding of 32-characters keys when it is installed. To enable the use of the AES 256-bit keys, the JCE requires the Unlimited Jurisdiction Policy Files. Download these files from the appropriate vendor as a post-installation step of the agent. Because of US Export Compliance considerations in using the Unlimited Jurisdiction Policy Files, consult your legal department for more information.

**Follow these steps:**

1. Download the (JCE) Unlimited Strength Jurisdiction Policy Files:
   - Windows, Linux (x86 and x64), Solaris (SPARC and x86): Oracle
   - AIX, z/Linux, i5/OS: IBM
   - HP-UX: Referred to Oracle
2. Extract the local_policy.jar and US_export_policy.jar files from the downloaded archive (zip) file.
3. Copy the local_policy.jar and US_export_policy.jar files to the following location, overwriting the existing policy files:
   - On UNIX:

     ```
     installDir/jre/lib/security
     ```
   - On Windows:

     ```
     installDir\jre\lib\security
     ```
4. Use the keygen utility to enter the 64-character hexadecimal key to use in the AES encryption algorithm.

   The following example shows setting up the key and saving it to the default location:

   ```
   Keygen 0x000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F AES
   ```
5. Restart the agent.

## Job Failed When Blob Did Not Contain Any Data

**Valid on CA Workload Automation AE**

In a command job, you can specify standard output or standard error as a binary large object (blob). If the size of the standard output or standard error file is 0 after the job runs, the job fails. After the fix, a warning message is written in the job log and the job completes successfully.

## New Agent Parameters for Connecting with a JMX Console

When configuring the agent to connect with a JMX console, you can now specify the following new agent parameters for accessing the JMX console:

**management.jmx.security.user**

> (Optional) Specifies the user for accessing the JMX console. You can use this parameter together with management.jmx.security.password to ensure that only authorized users can view, monitor, and control the agent using the JMX console.

**management.jmx.security.password**

> (Optional) Specifies the password for the user name specified in the management.jmx.security.user parameter.

**Note:** The password must be encrypted. To encrypt a password, use the Password utility that is provided with the agent. For more information about the Password utility, see the *CA Workload Automation Agent for UNIX, Linux, or Windows Implementation Guide*.

## Using Environment Variables that Start with an Underscore

If a job uses an environment variable that starts with an underscore (_), the job does not run. After the fix, jobs with environment variables that start with an underscore run correctly.

## Malformed AFMs Caused Agent to Crash

AFMs are messages that are sent between the agent and the scheduling manager. In this issue, the agent failed to parse an invalid AFM, causing the agent to crash. After the fix, the agent rejects AFMs with malformed mandatory AFM fields.

# Cannot Resolve STDOUT or STDERR Environment Variables

If you define STDOUT and STDERR as agent-wide or manager-specific environment variables, the variables are not resolved in the job definition.

For example, on UNIX, the following JIL attributes are defined on CA Workload Automation AE:

```
std_out_file: $STDOUT
std_err_file: $STDERR
```

On Windows, the following JIL attributes are defined on CA Workload Automation AE:

```
std_out_file: %STDOUT%
std_err_file: %STDERR%
```

To allow the agent to resolve the environment variables, set the following parameter to true in the agentparm.txt file:

```
oscomponent.lookupcommand=true
```

In addition, on CA Workload Automation ESP Edition or CA Workload Automation DE, set the following parameter to false or disable it in the agentparm.txt file:

```
oscomponent.noforceprofile=false
```

On CA Workload Automation AE, set the following parameters to true in the agentparm.txt file:

```
oscomponent.noforceprofile=true
oscomponent.cmdprefix.force=true
oscomponent.profiles.src.delay=true
```

After you configure the agent, these variables are now resolved correctly in the job definition.

# Verifying the Specified Job Profiles Exist Before the Job Runs (UNIX only)

**Valid on CA Workload Automation AE**

In CA Workload Automation AE, the agent sources the profile before running the job. In 11.3, if the job profile specified in the job definition does not exist, the job fails without an appropriate error message. After the fix, you can specify the following agent parameter to force profile checking:

**oscomponent.profiles.src.verify**

> Specifies whether the agent verifies that the job profile specified in the job definition exists before the job runs. This parameter is applicable if oscomponent.profiles.src.delay is set to true.
>
> **true**
>
> > Indicates that the agent verifies that the job profile specified in the job definition exists before the job runs. If the specified job profile does not exist, the job returns a submission error indicating the reason for the error.
>
> **false**
>
> > Indicates that the agent does not verify that the job profile specified in the job definition exists before the job runs. If the specified job profile does not exist, the job fails without an appropriate error message.
>
> **Default:** false

**More information:**

Configure the Agent for auto_remote on UNIX (see page 82)

# Redirecting Standard Output and Error with Windows Interactive Jobs

If a Windows interactive job redirects standard output or error, the job runs once and then fails on subsequent runs. After the fix, you can run Windows interactive jobs that redirect standard output or error.

## Setting the Current Working Directory Using PWD or HOME Environment Variables (Windows only)

If the set the PWD or HOME environment variables in a Windows job, the current working directory is not affected. After the fix, you can set the current working directory using the PWD or HOME environment variables in a Windows job. To set the current working directory, the agent checks the PWD environment variable first and then checks the HOME environment variable.

## Memory Consumption of File Watcher Jobs

**Valid on CA Workload Automation AE**

In r11.3, each File Watcher job started in its own JVM. As a result, each job consumed up to 512 MB of memory depending on the platform and machine configuration. After the fix, each File Watcher job consumes about 1 MB of memory. As a result, you can now run a lot more File Watcher jobs in parallel without running out of memory.

## Using Glob Regular Expressions in File Watcher Jobs

**Valid on CA Workload Automation AE**

In r11.3, File Watcher jobs do not support global binary large object (glob) regular expressions on UNIX. After the fix, you can use glob regular expressions with File Watcher jobs on UNIX. On Windows, you can use the * (match any sequence of characters) and ? (match any single character) wildcards with File Watcher jobs.

## Agent Did Not Send the Failed Keyword When a SUBERROR Occurred (Windows only)

When a Windows job with an execution user has a submission error (SUBERROR), the agent does not send the Failed keyword in the AFM. For example, as the result of this issue, an external job was mistakenly completed when its external dependency had a SUBERROR. After the fix, the agent sends the Failed keyword when a SUBERROR occurs.

# Job Failed When Command Name and Arguments are Enclosed in Quotes (Windows only)

In the job definition, when the command name and arguments are enclosed in quotes, as in the following example, the job fails:

```
"C:\Program files\test.bat"  "20"   " In Progress"
```

The issue occurs when oscomponent.cmdprefix.force is set to true, causing the agent to prefix cmd.exe for every Windows job.

To address this issue, an additional parameter has been added to the agentparm.txt file:

**oscomponent.cmdprefix.force.quotes.full**

Specifies whether the agent wraps the entire command in double quotes before the Windows command interpreter (cmd.exe) runs the command. This parameter is applicable if oscomponent.cmdprefix.force is set to true.

**true**

Indicates that the agent wraps the entire command in double quotes before cmd.exe runs the command. Set this parameter to true to allow commands that have spaces in their path run without error, for example:

```
"C:\ Program Files (x86)\command.bat" "C:\ Program Files (x86)\input-file"
```

**Notes:**

- Commands that contain embedded blanks in their paths succeed, for example: "C:\Program Files\program.exe".
- Commands with arguments fail if the entire command, including arguments, is quoted, for example: "C:\tools\program.exe arg1 arg2".

**false**

Indicates that the agent does not wrap the entire command in double quotes before cmd.exe runs the command.

**Notes:**

- Commands that contain embedded blanks in their paths fail, for example: "C:\Program Files\program.exe".
- Commands with arguments succeed if the entire command, including arguments, is quoted and the path does not contain embedded spaces, for example: "C:\tools\program.exe arg1 arg2".

**Default:** false

After you configure the agent, the command runs correctly.

# Specifying the Order Profiles are Sourced (UNIX only)

In 11.3, the agent sources the job profile first before EWAGLOBALPROFILE (/etc/auto.profile). You can now specify the following agent parameter to control the order profiles are sourced in:

**oscomponent.profiles.src.order.global.first**

Indicates whether EWAGLOBALPROFILE (/etc/auto.profile) is sourced before or after the job profiles.

**true**

Indicates that EWAGLOBALPROFILE is sourced first before the job profile.

**false**

Indicates that EWAGLOBALPROFILE is sourced last after the job profile.

**Default:** false

**More information:**

Configure the Agent for auto_remote on UNIX (see page 82)

## Enabling Chained Commands in a Job (UNIX only)

Currently, you cannot specify chained commands in a job, for example:

```
bin/sleep 10; echo hello
```

You can now add the following agent parameter to control whether chained commands are supported:

**oscomponent.wrapper.exec.force**

Specifies whether the wrapper script the agent generates puts exec in front of the target script or binary. This parameter is applicable if oscomponent.cmdprefix.force is set to true.

**true**

Indicates that the wrapper script the agent generates puts exec in front of the target script or binary.

**false**

Indicates that the wrapper script the agent generates does not put exec in front of the target script or binary. As a result, you can specify chained commands in a job, for example, bin/sleep 10; echo hello.

**Note:** If this parameter is set to false, you cannot send a signal to the job.

**Default:** false

**More information:**

Configure the Agent for auto_remote on UNIX (see page 82)

## Java Core Files Created During the Install

This release resolves an issue where Java core files were created during the agent installation.

## SNMP Parameter Mistakenly Set in Agent Parameter File

This release resolves an issue where the management.snmp.host parameter was mistakenly set in the agentparm.txt file when the following property was set in the install properties file:

```
SNMP_MGMT_CONN=0
```

## Passwords and Encryption Keys Visible in Clear Text

In this release, the following debug file has been removed:

```
CA_Workload_Automation_System_Agent_R11.3_Install_IA.log
```

By removing this log file, passwords and encryption keys are no longer visible in clear text.

## Complex Profile Caused Job That Should Fail to Complete Successfully (UNIX only)

This release resolves an issue where a complex profile caused a UNIX job that should fail to complete successfully. In this issue, the specified job profile called a second profile. The job failed on Solaris and Linux as expected, but ran to success on AIX.

The issue occurred with the following settings in the agentparm.txt file:

```
oscomponent.cmpdprefix.force=true
oscomponent.defaultshell.force=true
oscomponent.noexitcode=256
oscomponent.noforceprofile=true
oscomponent.profiles.src.delay=true
```

## Jobs Failed With Spool File Reading Failure Errors (i5/OS only)

On V7R1, the agent sometimes reports the following error after running for some time:

```
Spool file reading failure. Unable to retrieve *SEVERITY return code.
```

In this issue, the defaultlog_agent.log file located in the agent installation directory contains the following error:

```
Exception reading joblog spool file: An error occurred on the system
```

When this error occurs, all subsequent i5/OS jobs will be reported as failed even if they ran successfully on the i5 computer.

## Jobs Temporarily Marked Complete After Being Canceled (i5/OS only)

This release resolves an issue where long running i5/OS jobs were temporarily marked as complete after being canceled from the scheduling manager.

## CRYPTOJ SELF CHECK FAILED Errors

This release resolves "CRYPTOJ SELF CHECK FAILED" errors that occurred during agent installation and agent startup.

## Process Tree Not Terminated (Windows only)

This release resolves an issue where the native 64-bit Windows agent failed to terminate the entire process tree after the job cancel command was issued against a Windows job.

## Bypassing User Access Control (Windows only)

This release lets the agent start jobs in the background (not interactive) that bypass User Access Control (UAC) in Windows.

To start jobs in the highest privileges mode, set the following parameter in the agentparm.txt file:

```
oscomponent.logon.elevated=true
```

## External File Monitoring Job Failure

This release resolves an issue where submitting an external file monitor with the following setting resulted in job failure:

```
oscomponent.initialworkingdirectory=script
```

## File Watcher Job Failure (HP-Itanium only)

This release resolves the File Watcher job failure issue on 64-bit HP-Itanium systems.

# Transferring Multiple Mainframe Files Using Wildcards in Data Set Name

This release adds support for transferring multiple files on the mainframe using FTP jobs with wildcards (*) anywhere in the dataset name.

### Example

```
FTP_JOB CYBJK.FTP_DSN
  LOCALNAME 'C:\MF\CYB.FTP_DSN\DSN'
  REMOTENAME   '''KOLYE0*.LOG.MISC'''
  AFTER CYBJK.FTP_PDS
  RUN DAILY
ENDJOB

FTP_JOB CYBJK.FTP_DSW
  LOCALNAME 'C:\Tem\CYBJK.FTP_DSW'
  REMOTENAME '''CYBJAK2.*.I*C*'''
  AFTER CYBJK.FTP_DSN
  RUN DAILY
ENDJOB
```

On Linux, set the following parameter in the agentparm.txt file:

```
ftp.passive=true
```

# Extraneous Blank Lines Inserted When Transferring Large Files

This release resolves an issue where FTP jobs inserted extraneous blank lines when transferring large files.

# Reusing an Inactive Terminal Session

This release adds the ability to reuse the terminal session even if it is inactive.

To reuse an inactive terminal session, set the following parameter in the agentparm.txt file:

```
oscomponent.interactive.session.ignoreinactive=true
```

## Backing Up Obsolete Third-Party Libraries

This release adds the ability to back up obsolete third-party libraries that have no corresponding substitute. This ability is required for the development of the Web Services agent plug-in.

## Using UNC Pathnames for Standard Output and Standard Error (Windows only)

This issue resolves an issue with using UNC pathnames to redirect standard input and standard error files.

## Using Greater Than Sign for Standard Output

This release resolves an issue with using the greater than sign (>) to redirect standard output files.

## Agent Installation Errors (i5/OS only)

This release resolves errors that were generated during the installation on i5/OS.

## Agent Failed to Detect Second Intervention Request (i5/OS only)

This release resolves an issue where the agent did not detect the second intervention request and the job remained in an executing state.

# Appending to the Destination File During FTP or SFTP

This release enhances FTP and SFTP to allow the destination file to be appended instead of overwritten.

To append to the destination file during FTP or SFTP, add the >> characters before the destination file in the job definition, for example:

```
'>>/tmp/a2'
'>>c:\temp\ftp\tst1.txt'
```

**Notes:**

- On CA Workload Automation ESP Edition, use the following format if the destination file is a z/OS data set:

  ```
  '>>''HLQ.DSNAME(MEMBER)'''
  ```

- On UNIX, if the agent runs as root and performs FTP/SFTP download with append, the agent sets the owner of the file to the user specified in the LOCALUSER statement, if present.

# Inactive File Monitors

For inactive file monitors, this release adds support for newly discovered files to be handled in CREATE only fashion.

# Submitting Job with Empty Password Did Not Launch Correctly (Windows only)

This release resolves an issue where submitting a job with an empty password did not launch correctly.

# Keygen Utility Unable to Encrypt Short Passphrases

This release resolves an issue where the keygen utility was unable to encrypt passphrases that are 16 characters or fewer.

# SEND_SIGNAL Event Did Not Work for Certain Signals

This release resolves an issue where the SEND_SIGNAL event did not work for signals 1, 3, 9, and 15.

## Job Created Duplicate Standard Output Files When Variables Were Used

This release resolves an issue where the agent created duplicate files when a job is defined with an environment variable in its standard output filename.

## Cleaning Up Job Logs and Spool Files Automatically

This release alters job logs and spool files to be automatically cleaned up by the agent.

## Agent Sent Redundant Status Messages to the Scheduling Manager (UNIX only)

This release resolves an issue where the agent sent redundant status messages to the scheduling manager.

## cybAgent Binary Did Not Start (Solaris only)

This release resolves an issue where the cybAgent binary did not start on some Solaris versions because of a log file I/O problem.

## Arguments Not Passed to Job Because of Percent Sign (Windows only)

This release resolves an issue where arguments were not correctly passed to a job because the arguments included a percent sign (%) character.

## Autoping Failed When Primary Application Server Went Down in HADS Mode

This release resolves an issue where autoping failed when the primary application server went down in high availability dual event server mode.

## Object Monitors Failed Because of Race Condition

This release adds better error handling to prevent a race condition that caused object monitors to fail.

## File Trigger Status Not Updated Because of Special Character in Filename

This release resolves an issue where the agent could not update the scheduling manager of the file trigger status because the filename contained a special character.

## File Trigger Job Failed When Monitoring New Members in New Files (i5/OS only)

This release resolves an issue with using the GENERATE and recursive options to monitor new members in new files. In this issue, the File Trigger job failed if the job was not present.

## File Trigger Job Failed Because of Spaces in the Filename

This release resolves an issue where File Trigger jobs defined with spaces in the filename failed.

## File Triggers  Incorrectly Handled Filenames Enclosed in Double Quotes

This release resolves an issue in how file triggers handled filenames enclosed in double quotes.

## Incorrect AFM Syntax for File Monitors

This release resolves an issue with the file monitor AFM syntax.

## Incorrect Interpretation of Double Backslashes Prefixing UNC Path Names

This release resolves an issue with the interpretation of double backslashes prefixing UNC path names.

## Delays in Communication Between Agent and Scheduling Manager

This release improves handling of agent-scheduling manager communications to avoid delays.

## Job Goes into WAITFORNITIATOR Status When Unknown Job Class is Used

This release resolves an issue where a job went into WAITFORINITIATOR status when an unknown job class was used.

## File Trigger Jobs Failed with Condition Code 2

This release resolves an issue where File Trigger jobs failed with code 2 when trying to execute Java with the following setting:

```
filemonplugin.runexternal=true
```

## Leaked Initiators Canceling External File Monitors

This release resolves an issue where canceling an external file monitor resulted in a leaked initiator.

# Job Owner with C Shell as the Default Shell

The release resolves an issue when the default shell of the job owner is the C shell while allowing the agent to emulate the legacy auto_remote agent.

By default, the r11.3 SP1 agent acts differently than auto_remote for sourcing job profiles and global profiles from within a script when a user’s profile sourcing is disabled. You can configure agent parameters so that the r11.3 SP1 agent acts similar to auto_remote.

When you set the following parameters in the agentparm.txt file, the agent executes a temporary shell script without sourcing the user's profile:

```
oscomponent.noforceprofile=true
oscomponent.cmdprefix.force=true
```

When you set the following parameter, the agent drops the sourcing of the profiles into the temporary shell script:

```
oscomponent.profiles.src.delay=true
```

When you set the following parameter, additional rules apply:

```
oscomponent.profiles.global.override=true
```

- If the job profile is present, the agent sources the job profile. The EWAGLOBALPROFILE (/etc/auto.profile) is not sourced.
- If the job profile is not present, the agent sources EWAGLOBALPROFILE.

# Using Job Profiles for External File Monitors

This release adds an ability to use job profiles for submitting externally file monitors.

# SNMP Traps on Mixed IPv4/IPv6 Systems

This release resolves an issue sending SNMP traps by the management connector on mixed IPv4/IPv6 systems.

# File System Check Issue

This release resolves a file system check issue with paths containing spaces.

# Generating Temporary Shell Script (UNIX only)

If the following parameter is set, the agent generates a temporary shell script for UNIX command jobs:

```
oscomponent.cmdprefix.force=true
```

The temporary script is generated in the spool directory. The name of the temporary script is the same as the job log, except that it has an .sh extension instead of the .joblog extension.

The temporary script is erased when the spool file is erased. If the agent is configured to erase spool files automatically upon successful execution, the temporary script is also erased.

**Note:** The automatic erasure of the job log upon successful execution does not erase the temporary script.

# File Monitor Unable to Detect Large Files

This release uses stat64 to let the file monitor detect files with size greater than 2 GB minus 1 byte.

# Environment Variables in STDIN/STDOUT/STDERR Paths

This issue resolves issues with environment variables in standard input, standard output, and standard error paths. In this issue, the paths were enclosed in double-quotes and contained commands such as date.

# Chapter 5: Supported Systems and Requirements

This section contains the following topics:

## UNIX and Linux Platforms

CA WA Agent for UNIX or Linux r11.3 SP1 supports the following platforms:

**Notes:**

- These operating environments have been certified at the time of General Availability (GA). Additional operating environments may be certified post GA. For current information regarding operating environment support, check the CA Workload Automation Agent Product page at http://ca.com/support and follow the CA Workload Automation Compatibility Information link under the Product Status section.
- The agent has native 64-bit support on HP-Itanium and Linux.
- You can run a 32-bit agent on a 64-bit operating system, as long as the 32-bit libraries are installed.

| Platform | Supported Versions | Architecture | Notes |
|---|---|---|---|
| AIX | 5.3<br>6.1<br>7.1 | 32/64-bit | On AIX 6.1, install the latest fix pack from IBM.<br>See http://www-933.ibm.com/support/fixcentral |
| HP-UX | 11i v2 or 11i v3 | PA-RISC 32/64-bit | |
| HP-Itanium | 11i v3 | 64-bit | |
| Linux | RHEL 4, 5, or 6 | X64 or x86 32/64-bit, | You must install the |

| Platform | Supported Versions | Architecture | Notes |
|---|---|---|---|
| | SUSE Linux Enterprise Server 9, 10, or 11 | glibc-2.4<br>32/64-bit,<br>glibc-2.4 | compat-libstdc++ package (RPM) before installing the agent on Linux. |
| Solaris | 8, 9, 10, or 11 | SPARC 32/64-bit<br>x86 32-bit | |
| z/Linux | RHEL 4,5, or 6<br>SUSE Linux Enterprise Server 9, 10, or 11 | 31/64-bit,<br>glibc-2.4<br>31/64-bit,<br>glibc-2.4 | You must install the compat-libstdc++ package (RPM) before installing the agent on z/Linux. |

## UNIX and Linux System Requirements

The following table lists the minimum and recommended hardware requirements and software requirements for the agent.

**Notes:**

- Based on your workload volume and environment, you can require additional disk space.
- A Java Runtime Environment (JRE) runs the agent. The JRE is supplied with the agent for all platforms except z/Linux. For z/Linux, install the required JRE version.

**Important!** Do not replace the JRE that comes with the agent.

| Platform | Minimum Disk Space Required | Minimum Temporary Disk Space for Installation | JRE Version | Notes |
|---|---|---|---|---|
| AIX | 300 MB | 500 MB | JRE 6 update 10 | |
| HP-UX | 300 MB | 500 MB | JRE 6 update 14 | |
| Solaris | 300 MB | 500 MB | JRE 1.6 update 31 | |
| Linux | 300 MB | 500 MB | JRE 1.6 update 31 | |
| z/Linux | | | JRE 1.6 SR8, or higher<br>(31-bit) | The JRE is not supplied with the agent. |

# Windows Platforms

CA WA Agent for Windows r11.3 SP1 supports the following Windows platforms:

**Notes:**

- These operating environments have been certified at the time of General Availability (GA). Additional operating environments may be certified post GA. For current information regarding operating environment support, check the CA Workload Automation Agent Product page at http://ca.com/support and follow the CA Workload Automation Compatibility Information link under the Product Status section.
- The agent has native 64-bit support on Windows.
- The native CA WA Agent for Windows r11.3 fully exploits 64-bit technology with optimal performance running on the 64-bit version of the operating system. Earlier releases of the agent running 32-bit mode, while supported, have inherent limitations on 64-bit Windows. We recommend, if running the agent on the Windows 64-bit version of the operating system, that you match the native agent to run in 64-bit mode.

| Platform | Supported Versions | Architecture |
|---|---|---|
| Windows | 2003 R2 Server<br>2003 SP1<br>2008 R2 SP2<br>2008<br>XP SP3 Professional<br>Vista<br>7 | x86 32/64-bit |

## Windows System Requirements

The following table lists the minimum and recommended hardware requirements and software requirements for the agent.

**Notes:**

- Based on your workload volume and environment, you can require additional disk space.
- A Java Runtime Environment (JRE) runs the agent. The JRE is supplied with the agent.

**Important!** Do not replace the JRE that comes with the agent.

| Platform | Minimum Disk Space Required | Minimum Temporary Disk Space for Installation | JRE Version |
|---|---|---|---|
| Windows | 300 MB | 500 MB | JRE 1.6 update 31 |

# i5/OS Platforms

The agent supports any i5/OS or i5 system that supports i5/OS, Version V5R4M0, or higher.

**Note:** Ensure that IBM PTF SI27705 is installed on V5R4MO systems.

## i5/OS System Requirements

CA Workload Automation Agent for i5/OS requires the following environments:

- J2SE 5.0 32-bit (5722-JV1, Option 8)
- PASE (5722SS1 - Portable Application Solutions Environment, option 33)
- TCP/IP (5722-TC1) or TCP/IP (5722-AC1, AC2 or AC3) if you are using the agent to run SSL FTP workload
- Installation of the latest i5/OS CUM distribution
- The required group PTF levels for your i5/OS system

  For V5R4, the required minimum group PTF levels are as follows:

| Group PTF | Level | Description |
|---|---|---|
| SF99540 | 9321 | CUMULATIVE PTF PACKAGE C9321540 |

| Group PTF | Level | Description |
|---|---|---|
| SF99539 | 118 | Group Hiper PTF |
| SF99291 | 22 | Java Group PTF |
| SF99315 | 13 | TCP/IP Group PTF |

For V6R1, the required minimum group PTF levels are as follows:

| Group PTF | Level | Description |
|---|---|---|
| SF99610 | 10047 | CUMULATIVE PTF PACKAGE C0047610 |
| SF99609 | 57 | Group Hiper PTF |
| SF99562 | 11 | Java Group PTF |
| SF99354 | 5 | TCP/IP Group PTF |

For V7R1, the required minimum group PTF levels are as follows:

| Group PTF | Level | Description |
|---|---|---|
| SF99710 | 11116 | CUMULATIVE PTF PACKAGE C1116710 |
| SF99709 | 46 | Group Hiper PTF |
| SF99572 | 6 | Java Group PTF |
| SF99367 | 5 | TCP/IP Group PTF |

# Chapter 6: Related Documentation

This chapter includes documentation updates made as a direct result of the functionality changes and fixed issues in this service pack.

**Note:** CA Workload Automation Agent for UNIX, Linux, Windows, or i5/OS r11.3 SP1 includes the Release Notes only. This chapter refers to and should be read in conjunction with the CA Workload Automation Agent for UNIX, Linux, Windows r11.3 and CA Workload Automation Agent for i5/OS r11.3 documentation sets.

This section contains the following topics:

# How to Install and Configure the Agent

This topic was updated:

You can install the agent using an interactive program or using a command-based silent installer. If you are installing multiple agents, the silent installer lets you automate the installation process. After you install the agent, you can configure it to change your settings or to implement additional features. You also set up security features after the agent is installed.

**Important!** If you are installing the agent for use with CA Workload Automation AE, we recommend that you follow the directions in the *CA Workload Automation AE UNIX Implementation Guide* or *CA Workload Automation AE Windows Implementation Guide*. These guides refer to scripts that configure the agent specifically for use with CA Workload Automation AE.

To install and configure the agent, follow these steps:

1. Review the system requirements in the *CA Workload Automation Agent for UNIX, Linux, or Windows Release Notes*.
2. Collect information about the scheduling manager.
3. Review the agent installation program options (see page 67).
4. (z/Linux systems only)
   - Install the JRE (see page 67).
   - Set the PATH environment variable (see page 68).
5. Install the agent using one of these methods:
   - Install the agent on UNIX using an interactive program (see page 68) or install the agent on Windows using an interactive program (see page 69).
   - Install the agent using a silent installer (see page 70).
6. Configure the scheduling manager to work with the agent:
   - Define the agent on the scheduling manager.
   - (Optional) Define a user on the scheduling manager.
   - Configure security profiles on the scheduling manager.
   - Verify that the agent works with the scheduling manager.

   **Note:** For detailed instructions to complete these steps, see the documentation for your scheduling manager.
7. (Optional) Configure the agent.
8. Configure security features.

# Agent Installation Options

The Encryption Key installation option was updated:

**Encryption Key**

Defines the encryption key the agent uses to communicate with the scheduling manager. The encryption key must be prefixed with 0x and followed by the number of characters required for the chosen cipher algorithm:

- AES—32 hexadecimal character encryption key.

  **Note:** If you omit the 0x prefix, the keygen utility interprets the inputted value as a 16-character passphrase and not as a hexadecimal number. If you enter less than 16 characters, the keygen utility appends the passphrase with spaces for the missing number of characters. The keygen utility will internally encode the 16 character passphrase into a 32 hexadecimal character AES encryption key.

- Blowfish—32-64 even-numbered hexadecimal character encryption key
- DES—16 hexadecimal character encryption key
- DESEDE—48 hexadecimal character encryption key

The installation program provides the following default encryption keys that are based on the cipher algorithm you choose:

- DES—0xC468284E7BF6FB41
- DESEDE—0xB6F98D9B6419D51A41FEEEE847E27DC01F2B1ABE058142E3
- AES—0x32A26EA95ACC64B1B19DAF713D60DC10
- BLOWFISH—0xC832DD138ECCDC8A70F88E3120C6AA2F0148D725B19B3DF670B23E8514D6EC6B

**Note:** These keys were generated at random and have no set pattern.

# Install the JRE (z/Linux only)

This topic was updated:

If you are installing the agent on z/Linux systems, you must have the following Java Runtime Environment (JRE) version installed on your system:

JRE 1.6 SR8 or higher (The supported JRE is 31-bit. The 64-bit JRE is not supported.)

# Set the PATH Environment Variable (z/Linux only)

This topic was updated:

When you have the required JRE installed on your z/Linux system, you must set the PATH environment variable as follows:

```
export PATH=java_binary_location:$PATH
```

***java_binary_location***

Specifies the full path to the Java binary located in the JRE directory.

**Example:** export PATH=/usr/java6/jre/bin:$PATH

# Install the Agent on UNIX Using an Interactive Program

This topic was updated:

You can install the agent using an interactive program that lets you change and review your settings before starting the installation process. The installation program installs a packaged Java Virtual Machine (JVM) for the agent.

**To install the agent on UNIX using an interactive program**

1. Log on as root.
2. Copy the setup file from the product CD or download a zip file from the CA Support Online website, found at http://ca.com/support.
3. Copy or FTP the setup file to the target system and directory.
4. Type the following command to obtain execute permission for the setup file:

   ```
   chmod +x
   ```

5. (Optional) Set the IATEMPDIR environment variable to override the system temp directory:

   ```
   IATEMPDIR=/opt/CAWA/tempdir
   export IATEMPDIR
   ```

   ***tempdir***

   Specifies the path to a temporary directory the agent installation program uses during the installation process.

6. Type the following command to start the installation:

   ```
   ./setup.bin -i console
   ```

   The agent installation program opens.

7. Press Enter.

   The license agreement appears.

8. Type **y** to accept the license agreement.

9. Continue with the installation by entering the required information.

   **Notes:**

   - For z/Linux systems, you must have the required JRE installed and the PATH environment variable set to complete the installation.
   - To comply with U.S. Government encryption standard FIPS 140-2, select AES when you are prompted for the cipher algorithm.

10. Review your selections. To return to a previous option, type **back**.

11. Press Enter to exit the installation program.

    The agent is installed and the settings are stored in the agentparm.txt file located in the agent installation directory.

    **Notes:**

    - If you are installing the agent on a Linux computer that is SELinux enabled, a warning message appears. Change the default security context for IDL.
    - If you have problems with the agent installation, you can display debugging information for troubleshooting purposes.

# Install the Agent on Windows Using an Interactive Program

This topic was updated:

You can install the agent using an interactive wizard that lets you change and review your settings before starting the installation process.

**To install the agent on Windows**

1. Copy the setup file from the product CD or download a zip file from the CA Support Online website, found at http://ca.com/support.
2. Copy or FTP the setup file to the target system and directory.

3. Double-click **setup.exe**.

   The agent installation program opens.

4. Accept the license agreement and click Next.

   The Product Icons and Shortcuts dialog opens.

5. Continue with the installation by entering the required information.

   **Note:** To comply with U.S. Government encryption standard FIPS 140-2, select AES when you are prompted for the cipher algorithm.

   The Review Settings dialog appears as the last dialog before the installation process begins.

6. Review the settings and use the Back button to change the values you entered.

7. Click Install to begin the installation.

   The Monitor Progress dialog opens to show you the installation progress. The Installation Complete dialog opens when the installation process is finished.

8. Click Finish.

   The agent is installed and the settings are stored in the agentparm.txt file located in the agent installation directory.

   **Note:** If you have problems with the agent installation, you can display debugging information for troubleshooting purposes.

# How to Install the Agent Using a Silent Installer

This topic was updated:

A silent installer lets you automate the installation of multiple agents. You can configure a properties file for each agent and then run a silent installer instead of using an interactive program to install each agent.

**To install the agent using a silent installer**

1. Configure the installer properties file (see page 71).
2. Run the silent installer:
   - on UNIX (see page 74)
   - on Windows (see page 74)
3. Review the generated log file.

# Configure the Installer Properties File

This topic was updated:

You configure the installer properties file as the first step in performing a silent installation for one or more agents. We recommend that you keep a copy of this file to use as a template.

**To configure the installer properties file**

1. Open the installer properties file:
   - On UNIX:

     `unix_installer.properties`
   - On Windows:

     `win_installer.properties`

   The installer properties file is available on the product CD or CA Support Online website, found at http://ca.com/support.
2. Edit the properties for the agent. Remove the # sign to uncomment each property line.
3. Save the file.

   The properties are set in the installer properties file.

# Silent Installer Properties

The USER_SHORTCUTS property was removed.

The following properties were updated:

**JVM_DOT**

Specifies the full path to the JRE directory.

**Note:** This property is required for z/Linux systems only.

**Example:** /usr/java/jre1.6.0_16/jre specifies the path to the JRE directory for JRE 1.6.0_16.

**JVM_PATH**

Specifies the full path to the Java binary located in the JRE directory.

**Note:** This property is required for z/Linux systems only.

**Example:** /usr/java/jre1.6.0_16/jre/bin/java specifies the path to the Java binary location for JRE 1.6.0_16.

The following property was added:

**RAW_DATA**

Specifies the path to and name of a text file that contains additional agent parameters. In the text file, list each parameter on a separate line. During a silent installation, these parameters are inserted at the end of the agentparm.txt file without any validation or modification.

**Example:** /usr/home/joe/additional_agentparm.txt

# Silent Installer Example

This topic was updated:

The following example shows the property settings for installing an agent using the silent installer.

### Example: Configuring the Installer Properties File

The installer properties file in this example installs an agent named AGENT2 on a UNIX system in the agent_solaris_aes directory. The agent uses port 34520 for communication with the scheduling manager named manager1 that has an IP address of ::FFFF:192.168.00.00 and uses port 8507. Local security for AGENT2 is enabled. The agent uses the AES cipher algorithm. Environment variables used by the agent and scheduling manager are located in the FILE1.txt file. Additional parameters related to spool maintenance are appended at the end of the agentparm.txt file as listed in the additional_agentparm.txt file.

```
USER_INSTALL_DIR=/u1/build/CA/agent_solaris_aes
AGENT_INFO_1=AGENT2
AGENT_INFO_2=34520
NUM_MANAGER_1=1
MANAGER_1_INFO_1=manager1
MANAGER_1_INFO_2=::FFFF:192.168.00.00
MANAGER_1_INFO_3=8507
STRONG_ENCRYPTION_CIPHER=AES
STRONG_ENCRYPTION_KEYGEN=0x01020304050607080102030405060708
LOCAL_SECURITY=on
NUM_MANAGER_VARS_2=1
MANAGER_VARS_1_INFO_1=MANAGER1_VAR
MANAGER_VARS_1_INFO_2=C:\\MANAGER_1\\FILE1.TXT
NUM_USER_VARS_2=1
USER_VARS_1_INFO_1=USER1
USER_VARS_1_INFO_2=C:\\USER1\\FILE1.TXT
JOBLOG=true
RAW_DATA=/usr/home/joe/additional_agentparm.txt
```

The additional_agentparm.txt file specified in the RAW_DATA property contains the following:

```
runnerplugin.spool.clean.enable=true
runnerplugin.spool.expire=50000
runnerplugin.spool.sleep=20000
```

# Run the Silent Installer on UNIX

This topic was updated:

You can run the silent installer to perform the agent installation.

To run the silent installer on UNIX, type the following command at the command prompt:

```
./setup.bin -f path/unix_installer.properties
```

***path***

Specifies the full path to the installer properties file.

The agent is installed.

# Run the Silent Installer on Windows

This topic was updated:

You can run the silent installer to perform the agent installation.

To run the silent installer on Windows, type the following command at the command prompt:

```
setup.exe -f "path\win_installer.properties"
```

***path***

Specifies the full path to the installer properties file.

The agent is installed.

# Convert an Existing agentparm.txt File Using the Silent Installer

This topic was updated:

You can convert your Release 7 agentparm.txt file to r11.3 to preserve your existing settings. You can use the silent installer to do the conversion.

**To convert an existing agentparm.txt file using the silent installer**

1. Copy the r11.3 installer properties file to your agent computer.
   - On UNIX:

     ```
     unix_installer.properties
     ```

   - On Windows:

     ```
     win_installer.properties
     ```

   The installer properties file is available on the product CD or CA Support Online website, found at http://ca.com/support.
2. Open the installer properties file you copied.
3. Disable the following property by adding a comment (#) character:

   ```
   #AGENTPARM_CONVERT_2 =No
   ```

4. Enable the following property by removing the comment (#) character:

   ```
   AGENTPARM_CONVERT_1 =Yes
   ```

   Enabling this property preserves the agentparm.txt settings of your existing agent.
5. Enable and edit the following property to specify the path to the agentparm.txt file for your existing agent:

   ```
   #OLD_AGENT_PARM=C:\\Program Files\\Cybermation\\ESP System Agent
   R7\\agentparm.txt
   ```

6. Verify that all other properties are disabled by adding the comment (#) character to each property that is uncommented.
7. Save the installer properties file.
8. Run the silent installer:
   - on UNIX (see page 74)
   - on Windows (see page 74)
9. Review the generated log file.

# Uninstall the Agent on Windows

This topic was updated:

You can uninstall the agent after you have upgraded it from a previous release, or when you want to remove the agent from your system.

**To uninstall the agent on Windows**

1. Verify that all workload is complete.
2. Stop the agent.
3. Launch the uninstall program using Add/Remove Programs (Uninstall a program), similar to other Windows applications.

   The Uninstall CA Workload Automation Agent R11.3 dialog opens.
4. Click Uninstall.

   The product files are deleted and the agent is uninstalled.
5. Click Done to close the dialog.

# Configure the Agent for Windows Interactive Jobs

This topic was updated:

You can configure the agent that is installed on a Windows computer to submit jobs in interactive mode instead of in batch mode. Interactive mode lets users view and interact with jobs that invoke Windows Terminal Services or user interface processes, for example, Notepad.

**Note:** Not all scheduling managers support Windows Interactive jobs. Consult the *Release Notes* for your scheduling manager to determine whether this enhancement is supported.

**To configure the agent for Windows interactive jobs**

1. Change to the agent installation directory.
2. Stop the agent.
3. Open the agentparm.txt file.
4. Set the following parameter:

   ```
   oscomponent.interactive=true
   ```

   **Note:** By default, the agent runs interactive jobs using Terminal Services. To run interactive jobs using Terminal Services, the user that is specified to run the job must be logged in to the agent computer, either using Remote Desktop or locally. Also, the user must be logged in to the agent computer exactly once. If these conditions are not met during job submission, the job fails with a submission error. To overcome these limitations, run interactive jobs in Session 0.

5. (Optional) Define the following parameters to run interactive jobs in Session 0:

   **oscomponent.interactive.sessionzero**

   Sets whether the agent runs the interactive job in Session 0.

   **true**

   Runs all interactive jobs using Session 0. The user is not required to log in to the agent computer before the job runs. A greater amount of resources is available to run the job.

   **false**

   Runs all interactive jobs using Terminal Services.

   **Default:** false

**oscomponent.su.newconsole**

(Optional) Sets whether to force the agent to create a console when running a Windows job.

**true**

Creates a console when running a Windows job.

**false**

Does not create a console when running a Windows job.

**Default:** false

**oscomponent.cmdprefix.force.redir.inline**

(Optional) Sets whether the agent forces inline redirection. In some cases, such as Windows Server 2008 R2, inline redirection is required for the console to be displayed in Session 0.

**true**

Forces inline redirection using cmd.exe /c *command args stdin_file* >*stdout_file* >*stderr_file*.

**false**

Does not force inline redirection.

**Default:** false

**Note:** This parameter requires oscomponent.cmdprefix.force=true and affects FTP verbose mode. If regular redirection is used, the command ftp –s:command_file will not output raw FTP commands with completion codes as per FTP protocol.

**oscomponent.cmdprefix.force.redir.ifstdon**

(Optional) Sets whether to prevent the agent from redirecting STDOUT/STDERR if none are specified in the job definition.

**true**

Prevents the agent from redirecting STDOUT/STDERR if none are specified in the job definition. If the output of the command requires user input, such as pause, it is displayed in the cmd.exe console window.

**false**

Does not prevent the agent from redirecting STDOUT/STDERR if none are specified in the job definition.

**Default:** false

**Note:** This parameter requires oscomponent.cmdprefix.force=true and the Windows job has to be interactive.

**Note:** If you are not logged in when the job is submitted, you might need to restart the "Interactive Services Detection" service to receive the prompt.

6. (Optional) Define the following parameters:

   **oscomponent.shell**

   Specifies the executable file name for an alternative Windows shell.

   **Default:** explorer.exe

   **oscomponent.interactive.session.ignoreinactive**

   Sets whether the agent reuses an inactive terminal session.

   - false—Ignores an inactive terminal session. If the session is disconnected remotely, the interactive job fails.
   - true—Reuses an inactive terminal session.

   **Default:** false

7. Save and close the agentparm.txt file.

8. Start the agent.

   The agent is configured for Windows interactive jobs.

# Configure the Agent for Windows User Access Control

This topic was added:

You can configure an agent that is installed on a Windows computer to submit jobs in the highest privileges (elevated) mode. Elevated mode lets users bypass User Access Control (UAC). UAC improves the security of Microsoft Windows by limiting application software to standard user privileges until an administrator authorizes an elevation.

**To configure the agent for Windows User Access Control**

1. Change to the agent installation directory.
2. Stop the agent.
3. Open the agentparm.txt file.
4. Set the following parameter:

   **oscomponent.logon.elevated**

   Sets whether the agent submits jobs in elevated mode.

   - false—Does not submit the job in elevated mode on Windows.
   - true—Submits the job in elevated mode on Windows.

   **Default:** false

   **Notes:**

   - To schedule elevated jobs, the user specified to run the job must be logged in with an active session.
   - To schedule elevated jobs, the user can have only one active session. If more than one session exists for the specified user, the agent cannot determine which session to use and therefore fails the job.
5. Save and close the agentparm.txt file.
6. Start the agent.

   The agent is configured for Windows User Access Control.

# Register the Agent as a Windows Service

This topic was updated:

The agent must be registered as a Windows service before it can be started as a service. The agent installation program automatically registers the agent as a Windows service.

**Note:** You need the authority of an administrator or a server operator to start or stop services.

If you install more than one agent, register each agent as a service. Specify unique service and service display names in the agentparm.txt file for each agent.

**To register the agent as a Windows service**

1. Open the command prompt, and change to the agent installation directory.
2. Issue the following command:

   ```
   cybAgent -install
   ```

# Deregister the Agent as a Windows Service

This topic was updated:

When you deregister the agent as a service, it is stopped and then removed.

**To deregister the agent as a Windows service**

1. Open the command prompt, and change to the agent installation directory.
2. Issue the following command:

   ```
   cybAgent -remove
   ```

# Configure the Agent for auto_remote on UNIX

**Note:** This procedure applies to CA Workload Automation AE only. The CA Workload Automation AE documentation refers to auto_remote as the legacy agent.

CA Workload Automation Agent for UNIX, Linux, or Windows replaces the Remote Agent (auto_remote) that was provided with Unicenter AutoSys JM r4.5 and r11. By default, the agent behaves differently than the legacy agent for sourcing job profiles and global profiles, deleting spool files and job logs, and so on. You must manually add or edit the parameters in the agentparm.txt file to configure the agent to behave like the legacy agent.

**To configure the agent for auto_remote on UNIX**

1. Change to the agent installation directory.
2. Enter the following command:

   ```
   ./cybAgent -s
   ```

   The agent stops.
3. Open the agentparm.txt file and edit or add the following parameters to configure the agent, as follows:

   ```
   oscomponent.environment.variable=agent_installation_directory/profiles/WAAE.txt
   oscomponent.environment.variable_manager_instance_name_SCH=agent_installation_directory/profiles/instance_name.txt
   oscomponent.joblog.success.autocleanup=true
   agent.spool.success.autocleanup=true
   runnerplugin.spool.clean.enable=true
   runnerplugin.spool.expire=7d
   agent.resourcemon.enable=true
   filemon.firstscan.skip=true
   oscomponent.noexitcode=256
   oscomponent.auth.pam.svc=sshd
   security.cryptkey=cryptkey.txt
   oscomponent.cmdprefix.force=true
   oscomponent.noforceprofile=true
   oscomponent.profiles.src.delay=true
   oscomponent.profiles.src.order.global.first=true
   oscomponent.profiles.src.verify=true
   oscomponent.initialworkingdirectory=USER
   oscomponent.lookupcommand=true
   ```

   ***agent_installation_directory***

   Specifies the path to the directory where the agent is installed.

   ***instance_name***

   Specifies the name of the CA Workload Automation AE instance.

4. Save the agentparm.txt file.

5. Enter the following command:

   ```
   ./cybAgent -a
   ```

   The agent starts and is configured for auto_remote.

**Notes:**

- The oscomponent.environment.variable parameter is set to the location of the WAAE.txt file. The WAAE.txt file defines the environment settings for jobs started on behalf of all managers for all instances of CA Workload Automation AE. For more information about the WAAE.txt file and the environment variables, see the CA Workload Automation AE *Administration Guide* or the *User Guide*.

- The oscomponent.environment.variable_manager_*instance_name*_SCH parameter is set to the location of the *instance_name*.txt file. The *instance_name*.txt file includes the path to the auto.profile file, which is one of the several objects that source the environment for a job. For more information about the auto.profile file and how the environment for a job is sourced, see the *CA Workload Automation AE Administration Guide* or the *User Guide*.

- To run CA Workload Automation AE utilities as jobs, ensure that a CA Workload Automation AE client is installed on the computer where the agent is installed. If you have installed the agent on a computer where no other agents were installed previously, you must copy the WAAE.txt and *instance_name*.txt files from the computer where an agent is installed to the *agent_installation_directory*/profiles directory and then configure the parameters in the agentparm.txt file.

- When you set the following parameters, the agent executes a temporary shell script without sourcing the user's profile:

  ```
  oscomponent.noforceprofile=true
  oscomponent.cmdprefix.force=true
  ```

- When you set the following parameter, the agent drops the sourcing of the profiles into the temporary shell script:

  ```
  oscomponent.profiles.src.delay=true
  ```

- When you set the following parameter, the agent sources the UNIX global profile first prior to execution:

  ```
  oscomponent.profiles.src.order.global.first=true
  ```

- When you set the following parameter, the agent verifies the existence of a job profile prior to execution:

  ```
  oscomponent.profiles.src.verify=true
  ```

  If the job profile does not exist, the job terminates and returns an error.

# Configure the Agent for auto_remote on Windows

**Note:** This procedure applies to CA Workload Automation AE only. The CA Workload Automation AE documentation refers to auto_remote as the legacy agent.

CA Workload Automation Agent for UNIX, Linux, or Windows replaces the Remote Agent (auto_remote) that was provided with Unicenter AutoSys JM r4.5 and r11. By default, the agent behaves differently than the legacy agent for sourcing job profiles and global profiles, deleting spool files and job logs, and so on. You must manually add or edit the parameters in the agentparm.txt file to configure the agent to behave like the legacy agent.

**To configure the agent for auto_remote on Windows**

1. Change to the agent installation directory.
2. Stop the CA Workload Automation Agent (*MYAGENT*) service from the Windows Service Control Manager.

   ***MYAGENT***

   Specifies the name of the agent.
3. Open the agentparm.txt file and edit or add the following parameters to configure the agent, as follows:

   ```
   oscomponent.environment.variable=agent_installation_directory\Profiles\WAAE.t
   xt
   oscomponent.environment.variable_manager_instance_name_SCH=agent_installation
   _directory\Profiles\instance_name.txt
   oscomponent.joblog.success.autocleanup=true
   agent.spool.success.autocleanup=true
   runnerplugin.spool.clean.enable=true
   runnerplugin.spool.expire=7d
   agent.resourcemon.enable=true
   filemon.firstscan.skip=true
   oscomponent.noexitcode=256
   security.cryptkey=cryptkey.txt
   oscomponent.cmdprefix.force=true
   oscomponent.initialworkingdirectory=USER
   oscomponent.lookupcommand=true
   ```

   ***agent_installation_directory***

   Specifies the path to the directory where the agent is installed.

   ***instance_name***

   Specifies the name of the CA Workload Automation AE instance.
4. Save the agentparm.txt file.
5. Start the CA Workload Automation Agent (*MYAGENT*) service from the Windows Service Control Manager.

The agent starts and is configured for auto_remote.

**Notes:**

- The oscomponent.environment.variable parameter is set to the location of the WAAE.txt file. The WAAE.txt file defines the environment settings for jobs started on behalf of all managers for all instances of CA Workload Automation AE. For more information about the WAAE.txt file and the environment variables, see the *CA Workload Automation AE Administration Guide* or the *User Guide*.
- The oscomponent.environment.variable_manager_*instance_name*_SCH parameter is set to the location of the *instance_name*.txt file. The *instance_name*.txt file contains a set of environment variables for each CA Workload Automation AE instance. For more information about the environment variables and how the environment for a job is sourced, see the *CA Workload Automation AE User Guide*.
- To run CA Workload Automation AE utilities as jobs, ensure that a CA Workload Automation AE client is installed on the computer where the agent is installed. If you have installed the agent on a computer where no other agents were installed previously, you must copy the WAAE.txt and *instance_name*.txt files from the computer where an agent is installed to the *agent_installation_directory*/Profiles directory and then configure the parameters in the agentparm.txt file.

# Configure the Agent to Connect with a JMX Console

A JMX connector, built into the agent, lets you use a JMX console to monitor and control the agent. You can use any JMX console that implements JSR-160 to perform the following tasks on the agent:

- Discover metrics
- Query and modify values of various metrics
- Discover and invoke various functions
- Discover, subscribe, and receive notifications

To configure the agent to connect to a JMX console, configure the following agent parameters on the agent:

**management.connector_*n*=jmx**

Identifies the type of management connector the agent uses to connect to an external application, where n is an integer starting from 1.

Specify jmx to allow a JMX console to monitor and control the agent.

**management.jmx.host**

Specifies the host name or IP address where the JMX connector listens.

**management.jmx.port**

Specifies the port where the JMX connector listens.

Default: 1099

**management.jmx.security.user**

(Optional) Specifies the user for accessing the JMX console. You can use this parameter together with management.jmx.security.password to ensure that only authorized users can view, monitor, and control the agent using the JMX console.

**management.jmx.security.password**

(Optional) Specifies the password for the user name specified in the management.jmx.security.user parameter.

**Note:** The password must be encrypted. To encrypt a password, use the Password utility that is provided with the agent. For more information about the Password utility, see the *CA Workload Automation Agent for UNIX, Linux, or Windows Implementation Guide*.

# Set the Encryption on the Agent Using the Keygen Utility

You can install the agent with one of four types of encryption: AES, Blowfish, DES, or DESEDE. The encryption key is specified during the agent installation, but you can change it any time using this procedure.

The keygen utility provided with the agent lets you encrypt a key. By default, the encryption key is stored in the cryptkey.txt file located in the agent installation directory. You can replace the encryption key in this file or specify a different file to store it.

**Note:** Make a note of the encryption key, and set the same value on the scheduling manager.

**Follow these steps:**

1. Change to the agent installation directory.
2. Enter the following command at the command prompt:

   ```
   keygen 0xkey cipher destination
   ```

   ***key***

   Defines the encryption key the agent uses to communicate with the scheduling manager. The encryption key must be prefixed with 0x and followed by the number of characters required for the chosen cipher algorithm:

   - AES—32- or 64-character hexadecimal encryption key.

     **Note:** AES 256-bit encryption requires a 64-character hexadecimal key.

   - Blowfish—32 to 64 even-numbered character hexadecimal encryption key
   - DES—16-character hexadecimal encryption key
   - DESEDE—48-character hexadecimal encryption key

   **Limits:** 16-64 alphanumeric characters (any digits and letters A-F only)

   **Notes:**

   - CA Workload Automation AE and CA Workload Automation CA7 Edition support only AES encryption. Consult the documentation for your scheduling manager to determine which encryption types are supported.
   - If you omit the 0x prefix, the keygen utility interprets the inputted value as a 16-character passphrase and not as a hexadecimal number. If you enter less than 16 characters, the keygen utility appends the passphrase with spaces for the missing number of characters. The keygen utility will internally encode the 16-character passphrase into a 32-character hexadecimal character AES encryption key.

***cipher***

Specifies the type of cipher algorithm the agent uses to encrypt and decrypt messages sent to the scheduling manager. The agent supports the following types:

- AES—Advanced Encryption Standard that uses a 32- or 64-character encryption key. AES is the algorithm required by U.S. Government organizations to protect sensitive (unclassified) information (FIPS-140-2 compliance).
- BLOWFISH—A license-free encryption algorithm that uses an encryption key of 32 to 64 even-numbered characters.
- DES—Data Encryption Standard that uses a 16-character encryption key.
- DESEDE—Triple Data Encryption Algorithm that applies the DES algorithm three times to each data block.

**Default:** DES

**Note:** CA Workload Automation AE and CA Workload Automation CA7 Edition support only AES encryption. Consult the documentation for your scheduling manager to determine which encryption types are supported.

***destination***

(Optional) Specifies the name of a text file that stores the encryption key.

**Default:** cryptkey.txt

**Note:** If you specify a new text file, update the security.cryptkey parameter in the agentparm.txt file.

The keygen utility replaces the encryption key.

### Example: Encrypt a Key

This example encrypts the key 0x1020304050607080 for 16-character (DES) encryption:

```
keygen 0x1020304050607080 DES
```

## Problems Starting the Agent on AIX

This topic was deleted.

# Agent Will Not Start—cybAgent Script is Missing

This topic was updated:

If the cybAgent script is missing, you are unable to start the agent. The missing script can result when you perform a silent installation on z/Linux and specify incorrect values for the JVM_DOT or JVM_PATH variables. The installation program reports a successful installation even though these variables may be incorrect.

To verify whether this is the problem, check the CA_Workload_Automation_Agent_R11.3_InstallLog.log for the following text string:

```
Additional Notes: WARNING - Shortcut has no target: ignoring
```

After verification, correct the values of the JVM_DOT or JVM_PATH variables, and rerun the silent installer.

# Agent Parameters used for Troubleshooting

The following parameters were added:

**objmon.textmon.lines.upper.include**

Indicates whether to include the upper line number (TO) in the search range for Text Monitoring jobs.

**true**

Includes the upper line number (TO) in the search range (inclusive).

**false**

Excludes the upper line number (TO) from the search range (exclusive).

**Default:** false

**Note:** This parameter does not affect text searches that are based on date/time or regular expressions, which always include the upper boundary in the search.

**oscomponent.cmdprefix.force.quotes.full**

Specifies whether the agent wraps the entire command in double quotes before the Windows command interpreter (cmd.exe) runs the command. This parameter is applicable if oscomponent.cmdprefix.force is set to true.

**true**

Indicates that the agent wraps the entire command in double quotes before cmd.exe runs the command. Set this parameter to true to allow commands that have spaces in their path run without error, for example:

```
"C:\ Program Files (x86)\command.bat" "C:\ Program Files (x86)\input-file"
```

**Notes:**

- Commands that contain embedded blanks in their paths succeed, for example: "C:\Program Files\program.exe".
- Commands with arguments fail if the entire command, including arguments, is quoted, for example: "C:\tools\program.exe arg1 arg2".

**false**

Indicates that the agent does not wrap the entire command in double quotes before cmd.exe runs the command.

**Notes:**

- Commands that contain embedded blanks in their paths fail, for example: "C:\Program Files\program.exe".
- Commands with arguments succeed if the entire command, including arguments, is quoted and the path does not contain embedded spaces, for example: "C:\tools\program.exe arg1 arg2".

**Default:** false

**oscomponent.defaultfile.permission**

Specifies the standard UNIX file permission in octal notation starting with 0. The four-digit octal code specifies the default file access permissions for the following files that the agent creates:

- Temporary working shell scripts (CA Workload Automation AE only)
- Standard output and standard error files
- Job logs
- Spool files

**Example:** 0600 (grants read and write permissions to the owner, but prevents anybody else from accessing the file)

**Notes:**

- If oscomponent.defaultfile.permission is not specified, all files the agent creates will have the same permissions as before 11.3 SP1 cumulative 4.
- Temporary working shell scripts are granted execute permissions by the agent regardless of this parameter.
- This parameter does not change the access permission of the spool directory (that is, drwxrwxrwt).

**oscomponent.profiles.src.order.global.first**

Indicates whether EWAGLOBALPROFILE (/etc/auto.profile) is sourced before or after the job profiles.

**true**

Indicates that EWAGLOBALPROFILE is sourced first before the job profile.

**false**

Indicates that EWAGLOBALPROFILE is sourced last after the job profile.

**Default:** false

**oscomponent.profiles.src.verify**

Specifies whether the agent verifies that the job profile specified in the job definition exists before the job runs. This parameter is applicable if oscomponent.profiles.src.delay is set to true.

**true**

Indicates that the agent verifies that the job profile specified in the job definition exists before the job runs. If the specified job profile does not exist, the job returns a submission error indicating the reason for the error.

**false**

Indicates that the agent does not verify that the job profile specified in the job definition exists before the job runs. If the specified job profile does not exist, the job fails without an appropriate error message.

**Default:** false

**oscomponent.terminate.subtree**

Specifies whether the agent kills the process group or manually kills each process in the process subtree when a UNIX job is canceled.

- false—The agent sends the kill signal to the script's process group ID. As a result, the kill signal is delivered individually to all processes that are members of the group. Any process that is not a member of the group does not get terminated.
- true—The agent assembles the process subtree and manually kills each process. This approach ensures that all processes associated with the UNIX job are terminated.

**Default:** false

**Note:** If oscomponent.terminate.subtree is set to true, the agent sends the SIGKILL signal to cause the process to terminate immediately. The agent does not provide the option to send the SIGTERM signal, so the receiving process cannot perform any clean-up upon receiving the signal. The default behavior (oscomponent.terminate.subtree=false) does provide an option to send the SIGTERM signal.

**oscomponent.umask**

Provides support for the umask command, which turns off (disables) specific permissions that the oscomponent.default.permission parameter allows. The three-digit octal code sets the file mode creation mask (umask) for the following files that the agent creates:

- Temporary working shell scripts (CA Workload Automation AE only)
- Standard output and standard error files
- Job logs
- Spool files

**Example:** 066 (assuming the default file access permission is 666, this value turns off read and write permissions for the group and others)

**Notes:**

- If oscomponent.umask is not specified, the default umask of the user that started the agent is used for job logs, spool files, and wrapping scripts.
- For standard output and error files, the default umask of the user that runs the job is used with an exception on AIX and HP-UX. On AIX and HP-UX, the default umask is only used if the umask is set in the user profile.

**oscomponent.wrapper.exec.force**

Specifies whether the wrapper script the agent generates puts exec in front of the target script or binary. This parameter is applicable if oscomponent.cmdprefix.force is set to true.

**true**

Indicates that the wrapper script the agent generates puts exec in front of the target script or binary.

**false**

Indicates that the wrapper script the agent generates does not put exec in front of the target script or binary. Set this parameter to false to enable chained commands, for example, bin/sleep 10; echo hello.

**Note:** If this parameter is set to false, you cannot send a signal to the job.

**Default:** false

# Appendix A: Acknowledgements

This appendix contains copyright and license agreement information from third-party software used in CA Workload Automation.

This section contains the following topics:

## HP-UX JRE v.1.6

This Product is distributed with HP-UX JRE v.1.6. HP has provided additional copyright notices and information that may be applicable to portions of the HP-UX JRE in the THIRDPARTYLICENSEREADME.txt file that accompanies the HP-UX JRE files.

## IBM AIX JRE 6.0

Contains IBM Licensed Materials
Copyright IBM Corporation 2010

## Oracle JRE v.1.6

This Product is distributed with JRE v.1.6. Use of the Commercial Features of the JRE for any commercial or production purpose requires a separate license from Oracle. "Commercial Features" means those features identified Table 1-1 (Commercial Features In Java SE Product Editions) of the Software documentation accessible at http://www.oracle.com/technetwork/java/javase/documentation/index.html. Oracle has provided additional copyright notices and information that may be applicable to portions of the JRE in the THIRDPARTYLICENSEREADME.txt file that accompanies the JRE files and at http://www.oracle.com/technetwork/java/javase/documentation/index.html.