CA Workload Automation Agent for Remote Execution

Implementation Guide
r11.3.1, Second Edition
This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
This document references the following CA Technologies products:

- CA Process Automation
- CA Workload Automation AE
- CA Workload Automation Agent for Application Services (CA WA Agent for Application Services)
- CA Workload Automation Agent for Databases (CA WA Agent for Databases)
- CA Workload Automation Agent for i5/OS (CA WA Agent for i5/OS)
- CA Workload Automation Agent for Informatica (CA WA Agent for Informatica)
- CA Workload Automation Agent for Linux (CA WA Agent for Linux)
- CA Workload Automation Agent for Micro Focus (CA WA Agent for Micro Focus)
- CA Workload Automation Agent for Microsoft SQL Server (CA WA Agent for Microsoft SQL Server)
- CA Workload Automation Agent for Oracle E-Business Suite (CA WA Agent for Oracle E-Business Suite)
- CA Workload Automation Agent for PeopleSoft (CA WA Agent for PeopleSoft)
- CA Workload Automation Agent for Remote Execution (CA WA Agent for Remote Execution)
- CA Workload Automation Agent for SAP (CA WA Agent for SAP)
- CA Workload Automation Agent for UNIX (CA WA Agent for UNIX)
- CA Workload Automation Agent for Web Services (CA WA Agent for Web Services)
- CA Workload Automation Agent for Windows (CA WA Agent for Windows)
- CA Workload Automation CA 7 Edition
- CA Workload Automation DE
- CA Workload Automation Desktop Client (CA WA Desktop Client)
- CA Workload Automation ESP Edition
- CA Workload Control Center
Contact CA Technologies

Contact CA Support

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

- 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.

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.
Configuring an Alias on the Scheduling Manager .................................................................... 39

Chapter 6: Troubleshooting the Agent Plug-in ................................................................. 41
Job Goes in State SUBERROR Failed Due to Submission Problem........................................ 41
Job Fails Using FUP Command .......................................................................................... 42
Job Fails to Create Temporary Working File ..................................................................... 43
Job Fails Due to Long Command Line .............................................................................. 43

Chapter 7: Related Documentation .................................................................................. 45
CA Workload Automation AE Documentation ................................................................... 45
CA Workload Automation DE Documentation .................................................................... 46
CA Workload Automation ESP Edition Documentation .................................................... 46
CA Workload Automation CA 7 Edition Documentation .................................................. 47

Index ................................................................................................................................. 49
Chapter 1: Introduction

This section contains the following topics:

- Intended Audience (see page 7)
- Agents and Agent Plug-ins (see page 7)
- CA WA Agent for Remote Execution (see page 9)
- Job Types Supported by CA WA Agent for Remote Execution (see page 10)
- Properties Files (see page 10)

Intended Audience

This document is for system administrators who are responsible for upgrading, installing, and configuring agents.

You require knowledge of the operating system where the agent is installed and any third-party products or software technology that the agent uses. Before you run workload on a remote system, verify that you can log in to the remote system using Telnet or Secure Shell (SSH2).

Notes:

- The term Windows refers to any Microsoft Windows operating system supported by the agent.
- The UNIX instructions in this document also apply to Linux systems unless otherwise noted.

Agents and Agent Plug-ins

Agents are the key integration components of CA Technologies workload automation products. Agents let you automate, monitor, and manage workload on all major platforms, applications, and databases. To run workload on a particular system, you install an agent on that system. If your workload must run on a UNIX computer, for example, you can install and configure the CA WA Agent for UNIX. The agent can run UNIX scripts, execute UNIX commands, transfer files using FTP, monitor file activity on the agent computer, and perform many other tasks.
You can extend the functionality of the agent by installing one or more agent plug-ins in the agent installation directory. If you have a relational database such as Oracle, you can install a database agent plug-in to query and monitor the database. Other agent plug-ins are also available. For more information about agent plug-ins, see the Implementation Guide for the appropriate agent plug-in.

**Note:** The agent plug-ins are only available for UNIX, Linux, and Windows operating environments.

**Example: Workload with Different Types of Jobs**

The following workload contains z/OS jobs, a UNIX job, an SAP job, and a Windows job, running on different computers, in different locations, and at different times:

![Diagram of job workflow]

- **z/OS Job** runs daily in Toronto.
- **UNIX Job** runs daily in New York.
- **SAP Job** runs daily in Chicago.
- **Windows Job** runs daily in Chicago and weekly in Toronto.
- **z/OS Job** runs weekly in Toronto.
CA WA Agent for Remote Execution

The CA WA Agent for Remote Execution lets you connect to remote systems on UNIX, HP Integrity NonStop (Tandem), and OpenVMS. You can connect to the remote system using Telnet or Secure Shell (SSH2). With the CA WA Agent for Remote Execution, you can define and run remote execution jobs.

The CA WA Agent for Remote Execution lets a user perform the following tasks:

- Execute commands or scripts on a remote system
- Pass environment variables to the commands or scripts
- Verify the current job status of a given job
- Cancel a running job
- Read the job log
- Get the spool output for a given job

The following diagram shows the functional relationship between the scheduling manager, the CA WA Agent for Remote Execution plug-in, and the remote system:
Job Types Supported by CA WA Agent for Remote Execution

With the CA WA Agent for Remote Execution, you can define and run the following type of job:

Remote Execution

Lets you execute a script or a command on a remote system and pass arguments to it.

Properties Files

The agent uses two properties files to identify and communicate with a remote system: a system properties file and a custom properties file.

A system properties file defines default properties for the remote system such as the steps to take to log in. The agent is packaged with system properties files for each platform and protocol the agent supports. For example, you use the unixtelnet.properties file to connect the agent to a UNIX system using Telnet. You use the tandemssh2.properties file to connect the agent to an HP Integrity NonStop system using SSH2. You use the openvmsssh2.properties file to connect the agent to an OpenVMS system using SSH2. Each system properties file contains preset values for the properties, which you can configure as required by your system. These files are located in the config/proxy subdirectory of the agent installation directory after you install CA Workload Automation Agent for Remote Execution.

A custom properties file can contain mandatory properties that define the host name, connection port, and platform of the remote system, the communication protocol, the location for storing the agent spool file, and other properties. You create a custom properties file for each remote system you want the agent to connect to. The custom properties file can also contain optional properties to override properties that are specified in the system properties file.

More information:

Overriding System Properties (see page 29)
Chapter 2: Implementation Checklist

This section contains the following topics:

How to Install and Configure CA WA Agent for Remote Execution (see page 11)
Deciding Whether to Create an Alias (see page 12)

How to Install and Configure CA WA Agent for Remote Execution

CA WA Agent for Remote Execution is an agent plug-in that installs into the CA WA Agent for UNIX, Linux, or Windows installation directory.

Note: Before you install the agent plug-in, install and configure the agent.

To install and configure the agent plug-in, follow these steps:

1. Review the system requirements in the CA Workload Automation Agent for Remote Execution Release Notes.
2. Decide whether to create an alias (see page 12).
3. Apply the latest agent patch on the platform where you are installing the agent plug-in.
   For example, if you are installing on Windows, install the latest patch for CA Workload Automation Agent for Windows.
4. Install the agent plug-in using one of these methods:
   ■ Install using an interactive program (see page 13).
   ■ Install using a silent installer (see page 15).
5. Create a custom properties file for the remote system (see page 17).
6. (HP-UX remote systems only) Enable UNIX95 behavior (see page 20).
7. Configure the system properties for the remote system (see page 21), if necessary.
8. Configure the scheduling manager to work with the agent plug-in (see page 33).
9. (Optional) Run a verification test (see page 33).
Deciding Whether to Create an Alias

You install an agent plug-in into the agent installation directory to extend the core functionality of the agent. By default, the agent plug-in operates under the same agent name that is assigned to the agent. An alias lets you create a unique agent name for an agent plug-in, which is useful for controlling agent security or for setting up clustered environments. The installation program generates the alias by adding a suffix to the agent name.

**Note:** If you are installing the agent plug-in to work with CA Workload Automation DE, an alias is required. Setting up an alias for use with other scheduling managers is optional.

Suppose that you have installed an agent that is named AGT10 on a Windows computer. Users who have access to that agent through their security permissions specify AGT10 as the agent name in their job definitions. Now suppose that you install an agent plug-in into the installation directory for AGT10. You can create an alias for the agent plug-in, for example, AGT10_PROXY. You can then restrict access to that alias to only those users that run Remote Execution workload. Those specific users then must specify AGT10_PROXY as the agent name in their job definitions.

Each agent plug-in has a default alias that you can enable during installation. For example, the agent plug-in for CA WA Agent for Remote Execution has the default alias `agentname_PROXY`. You can enable or change the default alias name after installation. To work, also configure the alias on the scheduling manager.

**Note:** To configure agent aliasing for clustered environments, see the *CA Workload Automation Agent for UNIX, Linux, or Windows Implementation Guide*. 
Chapter 3: Installing the Agent Plug-in

This section contains the following topics:

- Controlling the Agent Plug-in (see page 13)
- Install the Agent Plug-in Using an Interactive Program (see page 13)
- Install the Agent Plug-in Using a Silent Installer (see page 15)
- Create a Custom Properties File for a Remote System (see page 17)
- System Properties (see page 21)
- Overriding System Properties (see page 29)
- How to Remove the Agent Plug-in (see page 30)

Controlling the Agent Plug-in

You control the agent plug-in using the agent where the plug-in is installed. Depending on your operating system, you have several options for starting or stopping the agent. On UNIX, issue a command to run a start or stop script. On Windows, start or stop the agent as a Windows service.

Note: For more information about starting and stopping the agent, see the CA Workload Automation Agent for UNIX, Linux, or Windows Implementation Guide.

Install the Agent Plug-in Using an Interactive Program

You can install CA WA Agent for Remote Execution using an interactive program that prompts you for the required information.

Follow these steps:

1. Copy the proxy.pak file into the agent installation directory. You can copy this file from the product DVD or can download a zip file that contains the file from the CA Support Online website (http://ca.com/support).

2. Change to the agent installation directory. For example, type the following command:
   - On UNIX:
     ```bash
cd opt/CA/WA_Agent_R11_3
     ```
   - On Windows:
     ```bash
cd C:\Program Files\CA\WA Agent R11.3
     ```
3. Stop the agent using one of the following commands:
   ■ On UNIX:
     ./cybAgent  -s
   ■ On Windows:
     cybAgent  -s

4. Start the installation program using one of the following commands:
   ■ On UNIX:
     ./PluginInstaller proxy.pak
   ■ On Windows:
     PluginInstaller.exe proxy.pak

   You are prompted to enable the default alias.

   **Note:** The default alias for CA WA Agent for Remote Execution is AGENT_PROXY.

5. Enter **Y** to enable the default alias or **N** to disable it.

   The installation program displays a message and closes automatically upon a successful installation.

6. Start the agent using one of the following commands:
   ■ On UNIX:
     ./cybAgent  &
   ■ On Windows:
     cybAgent  -a

   The agent starts.

   **Note:** The installation program backs up all modified and replaced files. The backup files are compressed into a file that is named backup_timestamp.zip, located in the backups subdirectory of the agent installation directory. You can use Winzip or other similar utilities to open the backup file. A backup copy of the agentparm.txt file is stored in the zip file.
Install the Agent Plug-in Using a Silent Installer

You can install CA WA Agent for Remote Execution using a silent installer to automate the installation. The silent installer requires an input file named silent.txt.

**Note:** This procedure assumes that you are not adding an alias. If you require an alias, we recommend that you install using an interactive program (see page 13).

**Follow these steps:**

1. **Change to the agent installation directory.** For example, type the following command:
   
   - **On UNIX:**
     
     ```bash
     cd opt/CA/WA_Agent_R11_3
     ```
   - **On Windows:**
     
     ```bash
     cd C:\Program Files\CA\WA Agent R11.3
     ```

2. **Stop the agent using one of the following commands:**
   
   - **On UNIX:**
     
     ```bash
     ./cybAgent -s
     ```
   - **On Windows:**
     
     ```bash
     cybAgent -s
     ```

3. **Run the silent installer using the following command:**

   ```bash
   PluginInstaller proxy.pak . path/silent.txt true
   ```

   **path**

   Specifies the path to the input file, silent.txt. If you do not specify a path, the silent.txt file must exist in the agent installation directory.

   **Note:** If you installed the agent plug-in using the interactive program, a silent.txt file is saved in the `install_dir/config/proxy` directory, where `install_dir` is the agent installation directory.

   The silent installer overwrites all files that are distributed during installation.

   **Note:** If you want to select which files to overwrite during installation, type the following command:

   ```bash
   PluginInstaller proxy.pak . path/silent.txt false
   ```

   Using false in the preceding command runs the silent installer as an interactive program. Modify the silent install input file (see page 16) when you use false; otherwise, the installation aborts.
4. Start the agent using one of the following commands:
   - On UNIX:
     
     ```
     ./cybAgent &
     ```
   - On Windows:
     
     ```
     cybAgent -a
     ```

**Modify the Silent Install Input File**

You can modify the silent install input file that is used to automate the installation or create a new file manually.

**Note:** Modify the silent install input file when you run the silent installer using the following command:

```
PluginInstaller proxy.pak . path/silent.txt false
```

**Follow these steps:**

1. Open the silent.txt file using a text editor.
   
   **Note:** The letter N on the first line of the file corresponds to the prompt asking whether to set up the alias. To simplify the silent installation, we recommend that you leave N.

2. Add one of the following letters on the next line of the file to indicate how you want to override files during the installation:
   
   - **Y**
     
     Overwrites one file.
   - **N**
     
     Does not overwrite one file.
   - **A**
     
     Overwrites all files.

   **Note:** For a simple installation, we recommend specifying A which overwrites all existing files.

3. Repeat the previous step for each of the remaining files if you specified Y or N in the previous step. Otherwise, you specified A and all of the remaining files will be overwritten.

4. Save the file.
Example: Modify the Silent Install Input File

In this example, the silent.txt file contains the following entries:

N
A

The following output appears when the silent installer runs using the preceding silent.txt file:

D:\Test\Agent\R11.3>PluginInstaller proxy.pak . silent.txt false
Do you want to set up the alias? (Y/N): N
Updating agent configuration file D:\Test\Agent\R11.3\agentparm.txt ...
Agent configuration file has been updated
D:\Test\Agent\R11.3\jars\proxy.jar exists. Override(Y/N/A)? A
Extracting D:\Test\Agent\R11.3\jars\proxy.jar ...
Extracting D:\Test\Agent\R11.3\jars\proxyjs.jar ...
Extracting D:\Test\Agent\R11.3\config\proxy\unixssh2.properties ...
Extracting D:\Test\Agent\R11.3\config\proxy\tandemssh2.properties ...
Extracting D:\Test\Agent\R11.3\config\proxy\tandemtelnet.properties ...
Extracting D:\Test\Agent\R11.3\config\proxy\openmsssh2.properties ...
Extracting D:\Test\Agent\R11.3\config\proxy\openmstelnet.properties ...
Extracting D:\Test\Agent\R11.3\config\proxy\CAWA-proxyAgent.license ...
Extracting D:\Test\Agent\R11.3\jars\sinetfactory.jar ...
Extracting D:\Test\Agent\R11.3\config\proxy\silent.txt ...
Extracting D:\Test\Agent\R11.3\config\proxy\sampleUnix.properties ...
Extracting D:\Test\Agent\R11.3\config\proxy\sampleTandem.properties ...
Extracting D:\Test\Agent\R11.3\config\proxy\sampleOpenvms.properties ...
Installer successful. Agent can now be restarted

Create a Custom Properties File for a Remote System

Create a unique custom properties file for each remote system you want the agent to connect to. You can use the agent to connect with up to six remote systems.

Note: The name that you give this file is required in a Remote Execution job definition to identify the remote system where the job runs.

Follow these steps:
1. Create a text file using a text editor.
2. Define the following properties in the text file:
   - host
     Specifies the IP address or host name of the remote system.
Create a Custom Properties File for a Remote System

port

Specifies the connection port of the remote system.

Default: 22 (SSH2 protocol) and 23 (Telnet protocol)

protocol

Specifies the type of protocol that is used for communication with the remote system. The following protocols are valid:

- ssh2
- telnet

Note: This value must be lowercase.

type

Specifies the name of the platform of the remote system. The following platforms are valid:

- tandem
- unix
- openvms

Notes:

- For Linux systems, specify unix as the type.
- This value must be lowercase.

spoolHome

Specifies the default location on the remote system where the agent stores job spool files.

Notes:

- The spoolHome directory must be readable and writeable.
- The user who executes the job must have read and write access to the spoolHome directory.
- The job log resides in the spool directory on the agent computer.
default.user

Specifies the default user name to connect to the remote system. The agent requires the default user name to verify the running job status. The default user requires the appropriate rights to read and remove files that other users create such as spool files.

Notes:

■ Verify whether the default user can log in to the remote system without having to respond to any messages. If the default user requires responses to messages to configure the environment, configure the login task that is specified in the system properties file (see page 21). Otherwise, the job will fail.

■ The agent does not support public key authentication for SSH. Specify a password (encrypted) for the default user in the default.password property.

default.password

Specifies the password corresponding to the default user.

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.

3. Save the file as target.properties in the config/proxy subdirectory of the agent installation directory.

target

Specifies a name for the remote system.

Limits: Up to 128 characters. You can use alphanumeric characters plus underscore (_). The first character must be alphabetic.

Note: Users must specify this name in a job definition to run a job on the remote system.

Example: Creating a Custom Properties File for a Remote UNIX System

In this example, the following properties are defined in the custom properties file for a remote system. The agent connects to a UNIX computer, named myunixhost, using Telnet. The agent uses the unixtelnet.properties system properties file for additional information that is required to log in to the remote system.

```properties
host=myunixhost
gtype=23
protocol=telnet
type=unix
spoolHome=/export/home/unixuser/spool
default.user=amusz
default.password=*****
```
Example: Creating a Custom Properties File for a Remote HP Integrity NonStop System

In this example, the following properties are defined in the custom properties file for a remote system. The agent connects to an HP Integrity NonStop computer, named tandemhost, using SSH2. The agent uses the tandemssh2.properties system properties file for additional information that is required to log in to the remote system.

```
host=tandemhost
port=22
protocol=ssh2
type=tandem
spoolHome=$DATA1.SPOOL
default.user=jlin
default.password=*****
```

Example: Creating a Custom Properties File for a Remote OpenVMS System

In this example, the following properties are defined in the custom properties file for a remote system. The agent connects to an OpenVMS computer, named openvmshost, using SSH2. The agent uses the openvmsssh2.properties system properties file for additional information that is required to log in to the remote system.

```
host=openvmshost
port=22
protocol=ssh2
type=openvms
default.user=pevbart1
default.password=B1B85DC9685809B7
spoolHome=dev$user:[pevbart1.spool]
```

More information:

Properties Files (see page 10)

Enable UNIX95 Behavior for Remote HP-UX Systems

To run a Remote Execution job on a remote HP-UX system, the agent requires that the following environment variables are set to enable UNIX95 behavior:

```
UNIX95=1
PS_PERSONALITY=unix95
```

To enable UNIX95 behavior, add a login task in the custom properties file for the remote HP-UX system.
Example: Enable UNIX95 Behavior Using SSH2

The following example enables UNIX95 behavior using SSH2:

```bash
loginTask.1.startPrompt=.*
loginTask.1.startPromptRegex=true
loginTask.1.endPrompt=.*
loginTask.1.endPromptRegex=true
loginTask.1.command=export UNIX95=1;PS_PERSONALITY=unix95
```

**Note:** If you add a login task to set the shell, it must precede this login task. In this example, you would use step number 1 to set the shell and step number 2 to enable UNIX95 behavior.

**More information:**

*Job Fails Due to Long Command Line* (see page 43)

---

**System Properties**

The agent uses the following properties to communicate with a remote system. You can change these values in the system properties file if most of your remote systems use the same values. If a remote system has unique values for these properties, add the properties to the custom properties file that you created for that remote system to override the default values.

**Note:** Do not add spaces at the end of the parameters. Verify that each line in your system properties file does not end with whitespace.

**completionCode**

Specifies the code that the agent uses to determine how a process ended. This property takes one of the following values:

- completion^procdeath:z^completion^code (specify to capture completion codes using PROCDEATH)
- COMPLETION:COMPLETIONCODE

You can define different completion codes for different users by defining a `username.completionCode` in the custom properties file.

**Note:** This property applies to HP Integrity NonStop systems only. This property is equivalent to the CompletionCode parameter on the legacy Telnet agent.
endPromptTimeout

Specifies the maximum time that is needed, in milliseconds (ms), to execute a command after a user successfully logs in.

Defaults:

- UNIX: 20000
- HP Integrity NonStop: 200000
- OpenVMS: 5000

fileTransfer.template

Specifies the format of the response that is returned by the job command that retrieves the location of the spool file. The spool file resides on the remote system.

The response format must include $PATH, which represents the path to the spool file. If the fileTransfer.template parameter is not set, the command returns the path to the spool file without formatting.

Example:


If the path to the spool file ($PATH) is /tmp/CAWA_spool/STRESS/MAIN/PROXY.1/GAR1.TXT, the command returns the following format:


Notes:

- Before you use FTP or HTTP formatting, set up and verify the FTP or HTTP setup on the remote system.
- For more information about retrieving the path to the spool file for a Remote Execution job, see your scheduling manager documentation.

lineTerminator

Specifies the line terminator that the system uses to send commands.

Defaults:

- UNIX: \n
- HP Integrity NonStop: \n\n
- OpenVMS: \n\n
**login.endPromptTimeout**

Specifies the maximum time that is needed, in milliseconds (ms), to execute every step of the login task before the system issues a timeout exception.

**Defaults:**

- UNIX: 40000
- HP Integrity NonStop: 50000
- OpenVMS: 5000

**Note:** This property is equivalent to the WaitFirstSysPrompt parameter on the legacy Telnet agent.

**loginTask.n.command**

Specifies the command that the agent inputs after it receives a specific prompt, where \( n \) is an integer that corresponds to the login step number. For example, when the agent receives the username prompt, the agent inputs the user name as the command.

**Note:** For HP Integrity NonStop, the agent only supports Tandem Advanced Command Language (TACL) commands.

**loginTask.n.endPrompt**

Specifies the end shell prompt this task should wait for to indicate that this task is finished, where \( n \) is an integer that corresponds to the login step number.

**loginTask.n.endPromptRegex**

Specifies the endPrompt as a regular expression, where \( n \) is an integer that corresponds to the login step number.

**loginTask.n.lineTerminator**

Specifies the line terminator that the system uses to send commands, where \( n \) is an integer that corresponds to the login step number.

**loginTask.n.startPrompt**

Specifies the shell prompt that this task should wait for before executing the command, where \( n \) is an integer that corresponds to the login step number.

**loginTask.n.startPromptRegex**

Specifies the startPrompt as a regular expression, where \( n \) is an integer that corresponds to the login step number.

**pathSeperator**

Specifies the platform dependant path separator.
promptTask.n.command

Specifies the command that the agent inputs after it receives a specific prompt, where n is an integer that corresponds to the login step number. For example, when the agent receives the username prompt, the agent inputs the user name as the command.

Defaults:
- UNIX: PS1='CA_PROMPT '
- HP Integrity NonStop: SETPROMPT NONE
- OpenVMS: SET PROMPT="CA_PROMPT "

promptTask.n.endPrompt

Specifies the end prompt this task should wait for to indicate that this task is finished, where n is an integer that corresponds to the login step number.

Defaults:
- UNIX: ((\[^'\])CA_PROMPT )|((?m)^CA_PROMPT )
- HP Integrity NonStop: >
- OpenVMS: CA_PROMPT

promptTask.n.endPromptRegex

Specifies the endPrompt as a regular expression, where n is an integer that corresponds to the login step number.

Defaults:
- UNIX: true
- HP Integrity NonStop: false
- OpenVMS: false

promptTask.n.lineTerminator

Specifies the line terminator that the system uses to send commands, where n is an integer that corresponds to the login step number.

Default: \r\n (HP Integrity NonStop)

promptTask.n.startPrompt

Specifies the prompt that this task should wait for before executing the command, where n is an integer that corresponds to the login step number.

Defaults:
- UNIX: .*
- HP Integrity NonStop: >
- OpenVMS: .*
**promptTask.\(n\).startPromptRegex**

Specifies the startPrompt as a regular expression, where \(n\) is an integer that corresponds to the login step number.

**Defaults:**

- UNIX: true
- HP Integrity NonStop: false
- OpenVMS: true

**Example: System Properties for Connecting to a UNIX System Using Telnet**

This example shows the system properties that are defined in the unix telnet.properties file for a UNIX system that the agent connects to using Telnet.

**Note:** Telnet requires a three-step login.

```plaintext
lineTerminator=\n
loginTask.1.startPrompt=login:
loginTask.1.startPromptRegex=false
loginTask.1.endPrompt=sword:
loginTask.1.endPromptRegex=false
loginTask.1.command={user.id}

loginTask.2.startPrompt=sword:
loginTask.2.startPromptRegex=false
loginTask.2.endPrompt=.*
loginTask.2.endPromptRegex=true
loginTask.2.command={user.password}

promptTask.1.startPrompt=.*
promptTask.1.startPromptRegex=true
promptTask.1.endPrompt=([\^\)]CA_PROMPT )|({?m)^CA_PROMPT )
promptTask.1.endPromptRegex=true
promptTask.1.command=PS1='CA_PROMPT '

login.endPromptTimeout=40000
endPromptTimeout=20000
...
```
Example: System Properties for Connecting to a UNIX System Using SSH2

This example shows the system properties that are defined in the unixssh2.properties file for a UNIX system that the agent connects to using SSH2.

**Note:** SSH2 requires a one-step login.

```
lineTerminator=\n
promptTask.1.startPrompt=.*
promptTask.1.startPromptRegex=true
promptTask.1.endPrompt=((^[']\r\n|\r\n|^m)CA_PROMPT )|((?m)^CA_PROMPT )
promptTask.1.endPromptRegex=true
promptTask.1.command=PS1='CA_PROMPT '

login.endPromptTimeout=40000
endPromptTimeout=20000
...
```

Example: System Properties for Connecting to an HP Integrity NonStop System Using Telnet

This example shows the system properties that are defined in the tandemtelnet.properties file for an HP Integrity NonStop system that the agent connects to using Telnet:

```
lineTerminator=\r\n
pathSeperator=.

loginTask.1.startPrompt=Enter Choice>
loginTask.1.startPromptRegex=false
loginTask.1.endPrompt=TACL 1>
loginTask.1.endPromptRegex=false
loginTask.1.lineTerminator=\n
loginTask.1.command=TACL

loginTask.2.startPrompt=TACL 1>
loginTask.2.startPromptRegex=false
loginTask.2.endPrompt=Password:
loginTask.2.endPromptRegex=false
loginTask.2.command=logon \{user.id\}
loginTask.2.lineTerminator=\r\n
loginTask.3.startPrompt=Password:
loginTask.3.startPromptRegex=false
loginTask.3.endPrompt=>
loginTask.3.endPromptRegex=false
loginTask.3.command=\{user.password\}
loginTask.3.lineTerminator=\r\n```
Example: System Properties for Connecting to an HP Integrity NonStop System Using SSH2

This example shows the system properties that are defined in the tandemssh2.properties file for an HP Integrity NonStop system that the agent connects to using SSH2:

```
promptTask.1.startPrompt=>
promptTask.1.startPromptRegex=false
promptTask.1.endPrompt=>
promptTask.1.endPromptRegex=false
promptTask.1.command=SETPROMPT NONE
promptTask.1.lineTerminator=\r\n
login.endPromptTimeout=50000
endPromptTimeout=200000
completionCode=\_COMPLETION:COMPLETIONCODE
...
```

Example: System Properties for Connecting to an OpenVMS System Using Telnet

This example shows the system properties that are defined in the openvmstelnet.properties file for an OpenVMS system that the agent connects to using Telnet.

```
loginTask.1.startPrompt=Username:
loginTask.1.startPromptRegex=false
loginTask.1.endPrompt=Password:
loginTask.1.endPromptRegex=false
loginTask.1.lineTerminator=\r\n
loginTask.1.command={user.id}
```
Example: System Properties for Connecting to an OpenVMS System Using SSH2

This example shows the system properties that are defined in the openvmsssh2.properties file for an OpenVMS system that the agent connects to using SSH2.

```
loginTask.2.startPrompt=Password:
loginTask.2.startPromptRegex=false
loginTask.2.endPrompt=.*
loginTask.2.endPromptRegex=true
loginTask.2.command={user.password}
loginTask.2.lineTerminator=\r\n
promptTask.1.startPrompt=.*
promptTask.1.startPromptRegex=true
promptTask.1.endPrompt=CA_PROMPT
promptTask.1.endPromptRegex=false
promptTask.1.command=SET PROMPT="CA_PROMPT "

login.endPromptTimeout=5000
endPromptTimeout=5000
...
```

More information:

Properties Files (see page 10)
Overriding System Properties

You can override a default property in the system properties file by defining the property in either the custom properties file or in the Remote Execution job definition.

A system property that is defined in the custom properties file overrides the value of the same property in the system properties file.

For some system properties, you can specify user-specific values by prefixing the property name with a user ID. The agent uses these values only when it connects and executes commands using the specified user. For example, if userid is specified in the job definition, then properties that are prefixed with userid override properties without the prefix.

In a Remote Execution job definition, you can specify a user to override the default.user and default.password properties that are specified in the custom properties file. You can also specify the full path to the spool file in a job definition to override the spoolHome property.

Example: Override a System Property in a Custom Properties File

Suppose that the system properties file contains the following property:

```
loginTask.3.endPrompt=\$.*> 
```

To override the system property on a remote system, the following property is set in the custom properties file:

```
loginTask.3.endPrompt=\${DATA userid} {d}*> 
```

In this example, the agent uses \${DATA userid} {d}*> for loginTask.3.endPrompt.

Example: Override a System Property for a Particular User

Suppose that the system properties file contains the following property:

```
loginTask.3.endPrompt=\$.*> 
```

To override the system property for a particular user, the following property is set in the custom properties file:

```
userid.loginTask.3.endPrompt=\${DATA userid} {d}*> 
```

If userid is specified in the job definition, the agent uses \${DATA userid} {d}*> for loginTask3.endPrompt.
Example: Set the System Prompt on the C Shell

Suppose that the system properties file contains the following property:

promptTask.1.command=PS1='CA_PROMPT '

To set the system prompt on the C shell (csh) for a particular user, the following property is set in the custom properties file:

user1.promptTask.1.command=set prompt='CA_PROMPT '

If user1 is specified in the job definition, the agent sets the system prompt on csh before running the job.

Example: Override the Spool File in a Job Definition

Suppose that the default spoolHome in the custom properties file is $CA1.JIM. To override the spool file, $CA1.JACK.MYSPOOL is specified as the spool file in the job definition. In this example, the generated spool file is stored under the $CA1.JACK subvolume and is named as MYSPOOL.

How to Remove the Agent Plug-in

You can remove an agent plug-in when you no longer require it.

To remove the agent plug-in, follow these steps:
1. Disable the agent plug-in (see page 30).
2. (Optional) Remove the agent plug-in from the scheduling manager (see page 32).

Disable CA WA Agent for Remote Execution

Use this procedure when you want to remove CA WA Agent for Remote Execution from your system.

Follow these steps:
1. Verify that all workload is complete.
2. Stop the agent.
3. Open the agentparm.txt file that is located in the agent installation directory.
4. Comment out the plugins.start_internal_\_n=proxy parameter.
5. Renumber all other agent plug-ins that are assigned a greater number than the agent plug-in you are uninstalling.
6. Comment out the communication.alias parameter if you created an alias during the agent plug-in installation.

7. Renumber any subsequent communication.alias_n parameters.

8. Save and close the agentparm.txt file.

9. Remove the proxy.jar and proxyjs.jar files from the jars subdirectory of the agent installation directory.

10. Start the agent.

**Example: Renumber the plugins.start_internal_n Parameter**

Suppose that you have the following agent plug-ins set in the agentparm.txt file:

```plaintext
plugins.start_internal_1=runner
plugins.start_internal_2=proxy
plugins.start_internal_3=ftp
plugins.start_internal_4=microfocus
```

To disable the agent plug-in for CA WA Agent for Remote Execution, you would modify the agentparm.txt file as follows:

```plaintext
plugins.start_internal_1=runner
#plugins.start_internal_2=proxy
plugins.start_internal_2=ftp
plugins.start_internal_3=microfocus
```

**Example: Renumber the communication.alias_n Parameter**

Suppose that you have two alias agent plug-ins. The agentparm.txt file has the following parameters:

```plaintext
communication.alias_1=AGENTNAME_PROXY
communication.alias_2=AGENTNAME_MF
```

To disable the agent plug-in for CA WA Agent for Remote Execution, modify the agentparm.txt file as follows:

```plaintext
#communication.alias_1=AGENTNAME_PROXY
communication.alias_1=AGENTNAME_MF
```
Remove the Agent Plug-in from the Scheduling Manager

In addition to disabling the agent plug-in, you can remove it from the configuration on the scheduling manager.

**Note:** For detailed instructions to remove the agent from the scheduling manager, see the documentation for your scheduling manager.
Chapter 4: Your Scheduling Manager and the Agent Plug-in

This section contains the following topics:

Configuring the Scheduling Manager to Work with the Agent Plug-in (see page 33)
Running a Verification Test (see page 33)

Configuring the Scheduling Manager to Work with the Agent Plug-in

If you have defined the following items for the agent plug-in, configure the items on the scheduling manager:

- An alias
- A user

For detailed configuration instructions, see the documentation for your scheduling manager.

Running a Verification Test

For your verification test, you define and run a Remote Execution job. To define the job, you require the following information:

Agent

Specifies the name of your agent or the alias for the agent plug-in if you created an alias.

Command

Specifies the command or script to run on the remote system. For example, you can try running the ls system command on UNIX or the DIR system command on HP Integrity NonStop or OpenVMS.

Limits: 1024 characters
**Target**

Specifies the name that is assigned to the remote system where the job should run. The name must match the name of the custom properties file for the remote system. For example, if the custom properties file is named unixbox.properties, then unixbox is the name that is required in the job definition.

**Limits:** 128 characters

For more information about defining a Remote Execution job, see the documentation for your scheduling manager.
Chapter 5: Configuring the Agent Plug-in

This section contains the following topics:

- How to Configure Agent Parameters (see page 35)
- CA WA Agent for Remote Execution Parameters in the agentparm.txt File (see page 36)
- How to Set Up an Alias for the Agent Plug-in (see page 38)

How to Configure Agent Parameters

You configure agent parameters by editing the agentparm.txt file, which is located in the agent installation directory. When you install the agent, the installation program adds frequently configured agent parameters to the file. Other agent parameters exist, which you must manually add to the agentparm.txt file to configure the agent. For any configuration changes to take effect, always stop and restart the agent. For some agent parameters, such as the agent name and communication parameters, also configure the parameters on the scheduling manager.

To configure agent parameters, do the following steps:

1. Configure agent parameters on the agent (see page 35).
2. Configure agent parameters on the scheduling manager (see page 36).

Configure Agent Parameters on the Agent

Configure agent parameters on CA WA Agent for UNIX, Linux, or Windows using the following procedure.

Follow these steps:

1. Change to the agent installation directory.
2. Stop the agent using one of the following commands:
   - On UNIX:
     
     .\cybAgent  
     
   - On Windows:
     
     cybAgent  

3. Open the agentparm.txt file that is located in the agent installation directory.
4. Make the required changes to the agent parameters.
5. Save and close the agentparm.txt file.
6. Start the agent using one of the following commands:
   - On UNIX:
     ```
     ./cybAgent &
     ```
   - On Windows:
     ```
     cybAgent -a
     ```

**Configure Agent Parameters on the Scheduling Manager**

When you change an agent parameter in the agentparm.txt file that is also defined on the scheduling manager, such as the agent name, configure the agent parameter on the scheduling manager.

**Note:** For detailed instructions to configure agent parameters on the scheduling manager, see the documentation for your scheduling manager.

**CA WA Agent for Remote Execution Parameters in the agentparm.txt File**

You can configure the following parameters in the agentparm.txt file. The file is located in the agent installation directory. You can open the agentparm.txt file in any standard text editor.

- **communication.alias_<n>**
  - Defines the alias name for the agent. The <n> suffix increments sequentially for each alias agent.
  - **Note:** To enable an alias on the agent, verify that the comment character (#) is removed from the parameter line.
  - **Default:** AGENT_PROXY

- **plugins.start_internal_<n>**
  - Specifies the agent plug-in to start by the core Java agent.
    - <n>
      - Denotes an integer that is assigned to the agent plug-in, starting at 1. The <n> suffix must increase sequentially for each agent plug-in.
proxy.connection.debug.enable

Sets whether debugging information is stored while the agent plug-in connects to a remote system. The output is stored in a log file that is located in the log subdirectory of the agent installation directory. The log file is named hostnamecommand.log, where hostname is the host name of the remote system.

true

   Enables debugging.
false

   Disables debugging.

Default: true

proxy.connection.retry.interval

Specifies the interval in seconds between attempts to connect to the remote system. If the Telnet or SSH2 server closes the connection (for example, due to network failure or too many concurrent connections), the agent attempts to reconnect after the specified interval. The proxy.ssh2.connection.retry and proxy.telnet.connection.retry parameters control the number of reconnection attempts.

Default: 30 seconds

proxy.enable.rawdata.log

Sets debugging for troubleshooting purposes under CA staff instruction.

Default: false

proxy.healthCheckMonitorInterval_ms

Specifies the running interval in milliseconds (ms) that the agent uses to check monitor threads for lockups (lack of response). If a lockup is detected (no response during the interval), the monitor thread is restarted.

Default: 1800000 (30 minutes)

proxy.jobmonitor.pollinterval_ms

Specifies how often in milliseconds (ms) the agent polls the Oracle Applications database for job status information.

Default: 10000 (10 seconds)
How to Set Up an Alias for the Agent Plug-in

During the agent plug-in installation, you are prompted to create a default alias, which you can change after installation. If you enable an alias on the agent plug-in, also configure the alias on the scheduling manager.

To set up an alias for the agent plug-in, follow these steps:

1. Create an alias for the agent plug-in (see page 39).
2. Configure the alias on the scheduling manager (see page 39).

proxy.maxSubmitConnections

Specifies the maximum number of active connections that can be used for job submission. The proxy.maxSubmitConnections parameter controls how many job requests can be submitted simultaneously. Before a Remote Execution job is submitted, it requires a separate connection to the remote system. When the maximum number of active connections is reached, subsequent job requests are queued until connections are freed up. A connection is freed up after a job is submitted.

Default: 4

Note: Do not change the default unless instructed by CA.

proxy.ssh2.connection.retry

Specifies the number of times the agent voluntarily retries the TCP connection using SSH2. The proxy.connection.retry.interval parameter controls how long the agent waits between reconnection attempts.

Default: 4

Note: Do not change the default unless instructed by CA.

proxy.telnet.connection.retry

Specifies the number of times the agent voluntarily retries the TCP connection using Telnet. The proxy.connection.retry.interval parameter controls how long the agent waits between reconnection attempts.

Default: 4

Note: Do not change the default unless instructed by CA.
Create an Alias for the Agent Plug-in

An alias lets you create a unique agent name for an agent plug-in. Each agent plug-in has a default alias, which you can enable or change.

To create an alias for the agent, configure the following parameter on the agent:

```
communication.alias_n
```

Defines the alias name for the agent. The \( n \) suffix increments sequentially for each alias agent.

**Note:** To enable an alias on the agent, verify that the comment character (\#) is removed from the parameter line.

**More information:**

Configure Agent Parameters on the Agent (see page 35)

Configuring an Alias on the Scheduling Manager

When you create an alias for an agent plug-in, also configure the alias on the scheduling manager. Define the alias on the scheduling manager with the same address, port number, and encryption key as the agent where the agent plug-in is installed.

**Note:** For detailed instructions to configure an alias on the scheduling manager, see the documentation for your scheduling manager.
Chapter 6: Troubleshooting the Agent Plug-in

This section contains the following topics:

- Job Goes in State SUBERROR Failed Due to Submission Problem (see page 41)
- Job Fails Using FUP Command (see page 42)
- Job Fails to Create Temporary Working File (see page 43)
- Job Fails Due to Long Command Line (see page 43)

Job Goes in State SUBERROR Failed Due to Submission Problem

Valid on all remote systems

Symptom:

When I run a Remote Execution job, the job goes in state SUBERROR Failed with an error message similar to the following:

```
/MAIN State SUBERROR Failed Status(Submission problem) LStatus("Logon failed due to timeout. Last response is ..You entered an invalid login name or password....
/MAIN State SUBERROR Failed Status(Submission problem) LStatus("Logon failed. Server response: Login incorrect") SetEnd Cmpc(1)
```

Solution:

The message indicates a connection problem.

To correct this problem, verify that the correct user and password are specified in the job definition and custom properties file for the agent.
Job Fails Using FUP Command

Valid on HP Integrity NonStop remote systems

Symptom:

When I run a Remote Execution job that executes an FUP command, the job fails and the spool file contains command abended.

Solution:

When the command is executing, a new process is launched in the background by the command. The new process tries to use a file that the agent process is using, so the two processes are competing for the same file. For this reason, the command abends.

To correct this problem, use a series of commands.

Example: Use a Series of Commands

The following example executes the FUP SUBVOLS command:

Command1("#PUSH in_variable out_variable zs ~; #SET in_variable SUBVOLS ~; $system.system.FUP /INV in_variable, OUTV out_variable, status zs ~; #wait zs")
Command2("#OUTPUTV out_variable ~; #POP in_variable out_variable zs")
Job Fails to Create Temporary Working File

Valid on HP Integrity NonStop remote systems

Symptom:

When I run a Remote Execution job, the job fails with a message similar to the following:

Failed to create temporary working file X13F because it already exists

Solution:

The message indicates that the agent cannot find an available temporary working file name under the spool file directory.

To correct this problem, purge the unused temporary working files under the spool file directory.

Note: The temporary files include spool files. To preserve the spool files, back up the temporary files before you purge them.

We recommend that you configure the agent to clear spool files automatically by setting the following parameter in the agentparm.txt file:

runnerplugin.spool.clean.enable=true

Note: For more information about configuring the agent to clear spool files automatically, see the CA Workload Automation Agent for UNIX, Linux, or Windows Implementation Guide.

Job Fails Due to Long Command Line

Valid on UNIX remote systems

Symptom:

When I run a Remote Execution job that contains a long command line, the job fails with status Invocation problem.

Solution:

Some shells restrict the number of characters on a line. If your job fails because the command line is too long, you can switch to a shell that supports invoking longer commands such as Bash.

To correct this problem, add a login task in the custom properties file that you are using for the remote system.
Example: Change the Shell Using Telnet

The following example changes the shell to Bash using Telnet:

```plaintext
loginTask.3.startPrompt=.*
loginTask.3.startPromptRegex=true
loginTask.3.endPrompt=.*
loginTask.3.endPromptRegex=true
loginTask.3.command=bash
```

**Note:** If you add a login task to enable UNIX95 behavior on HP-UX, this login task must precede it. In this example, you would use step number 3 to set the shell and step number 4 to enable UNIX95 behavior.

Example: Change the Shell Using SSH2

The following example changes the shell to Bash using SSH2:

```plaintext
loginTask.1.startPrompt=.*
loginTask.1.startPromptRegex=true
loginTask.1.endPrompt=.*
loginTask.1.endPromptRegex=true
loginTask.1.command=bash
```

**Note:** If you add a login task to enable UNIX95 behavior on HP-UX, this login task must precede it. In this example, you would use step number 1 to set the shell and step number 2 to enable UNIX95 behavior.

More information:

[Enable UNIX95 Behavior for Remote HP-UX Systems](#) (see page 20)
Chapter 7: Related Documentation

Documentation for the agent and scheduling managers is available in HTML and PDF format at [http://ca.com/support](http://ca.com/support).

**Note:** To view PDF files, download and install the Adobe Reader from the Adobe website if it is not already installed on your computer.

This section contains the following topics:

- [CA Workload Automation AE Documentation](#) (see page 45)
- [CA Workload Automation DE Documentation](#) (see page 46)
- [CA Workload Automation ESP Edition Documentation](#) (see page 46)
- [CA Workload Automation CA 7 Edition Documentation](#) (see page 47)

### CA Workload Automation AE Documentation

To work with the agent and CA Workload Automation AE, see the following documentation:

<table>
<thead>
<tr>
<th>Task</th>
<th>Documentation</th>
</tr>
</thead>
</table>
| Configure the scheduling manager to work with the agent | [CA Workload Automation AE UNIX Implementation Guide](#)  
[CA Workload Automation AE Windows Implementation Guide](#) |
| Define, monitor, and control jobs         | [CA Workload Automation AE Reference Guide](#)  
[CA Workload Automation AE User Guide](#)  
[CA Workload Control Center Workload Scheduling Guide](#) |
CA Workload Automation DE Documentation

To work with the agent and CA Workload Automation DE, see the following documentation:

<table>
<thead>
<tr>
<th>Task</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the scheduling manager to work with the agent</td>
<td>CA Workload Automation DE Admin Perspective Help</td>
</tr>
<tr>
<td>Define jobs</td>
<td>CA Workload Automation DE Define Perspective Help</td>
</tr>
<tr>
<td>Monitor and control jobs</td>
<td>CA Workload Automation DE Monitor Perspective Help</td>
</tr>
</tbody>
</table>

Note: The online help is available in HTML and PDF formats.

CA Workload Automation ESP Edition Documentation

To work with the agent and CA Workload Automation ESP Edition, see the following documentation:

<table>
<thead>
<tr>
<th>Task</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the agent to work with the scheduling manager</td>
<td>CA Workload Automation ESP Edition Installation and Configuration Guide</td>
</tr>
<tr>
<td>Monitor and control jobs</td>
<td>CA Workload Automation ESP Edition Operator’s Guide</td>
</tr>
</tbody>
</table>
To work with the agent and CA Workload Automation CA 7 Edition, see the following documentation:

<table>
<thead>
<tr>
<th>Task</th>
<th>Documentation</th>
</tr>
</thead>
</table>
| Configure the scheduling manager to work with the agent | CA Integrated Agent Services Implementation Guide  
CA Workload Automation CA 7 Edition Interface Reference Guide  
CA Workload Automation CA 7 Edition Systems Programming Guide |
| Define, monitor, and control jobs         | CA Integrated Agent Services User Guide  
CA Workload Automation CA 7 Edition Interface Reference Guide  
CA Workload Automation CA 7 Edition Database Maintenance Guide  
CA Workload Automation CA 7 Edition Command Reference Guide |
Index

A
agent plug-in
  configuring • 35
  controlling • 13
  description • 7
  function • 9
  installation and configuration process • 11
  installing (UNIX, Windows) • 13
  prerequisite knowledge • 7
  silent installation • 15
  uninstall process • 30
agentparm.txt file, parameters • 36
  alias
    setup process • 38

C
configuration parameters, descriptions • 36
configuration process, agent plug-in • 35
custom properties file
  creating • 17
  description • 10

D
documentation, scheduling manager • 45

I
installation
  automated • 15
  instructions • 13
  interactive • 13
  process • 11
  removing the plug-in • 30
  verification test • 33

J
job types, supported • 10

P
properties file
  description • 10
  remote system required properties • 17

R
remote system
  configure system properties • 29
  define mandatory properties • 17

S
scheduling manager
  configuration for the agent plug-in • 33
  configuring agent parameters • 36
  configuring an alias • 39
  documentation • 45
  silent installation, process • 15
  starting and stopping, agent plug-in • 13
  system properties
    descriptions • 21
    overriding • 29

T
tandemssh2.properties file, description • 10
troubleshooting, agent plug-in • 41

U
unixtelnet.properties file, description • 10

V
verification test, agent plug-in installation • 33

W
workload example, different jobs • 7