BROADCOM SOFTWARE

Workload Automation Agent for Remote Execution 24.0

Table of Contents

Release Information

The Release Information provides details on the latest releases, cumulative patches, and links to key resources, including release notes, the support matrix, build details, and TPSR.

Release Notes 24.0

Workload Automation Agent for Remote Execution 24.0 contains the following.

Enhancements

Workload Automation Agent for Remote Execution 24.0 includes the following enhancements.

Compatibility with Java 21

Workload Automation Agent for Remote Execution now supports Java 21. This update allows organizations to leverage Java's latest platform advancements. In addition, they can ensure the agent is using components actively supported by their vendors and up to date to reduce security vulnerabilities.

Support for Remote Agent on Windows Systems

Added support for the Remote Agent to connect to a remote Windows system and perform operations. For more information, see Install WA Agent for Remote Execution.

Changes to Existing Features

Workload Automation Agent for Remote Execution 24.0 includes no changes to existing features.

Fixed Issues

Workload Automation Agent for Remote Execution 24.0 includes no defect fixes.

Known Issues

There are no known issues reported in Workload Automation Agent for Remote Execution 24.0.

Related Topics

Workload Automation System Agent Release Notes.

Support Matrix

This matrix lists the versions of remote platforms that are supported by Workload Automation Agent for Remote Execution.

Considerations

Broadcom supports third-party vendor products only if both the Broadcom and third-party products are currently supported. If a third-party vendor discontinues support for their product, Broadcom will also discontinue support.

Support Matrix

Remote Execution	24.0	12.1
Microsoft Windows Server 2022	✓	#
AIX 7.3	✓	✓
AIX 7.2 TL5	✓	✓
HP Integrity NonStop J06.08 (Blade)	✓	✓
HP Integrity NonStop H06.20 (NS-Series)	✓	✓
HP Integrity NonStop G06.32 (S-Series)	✓	✓
HP OpenVMS 8.4	✓	✓
HP-UX 11i v3	✓	✓
Red Hat Enterprise Linux 9	✓	✓
Red Hat Enterprise Linux 8	✓	✓

Related Topics

Workload Automation Agent Platform Support Matrix

Plug-In Specific Parameters

Remote Systems Usage Information

Build Levels

At the time of General Availability (GA), the release and corresponding build number for Workload Automation Agent for Remote Execution are listed below.

Release	Build
24.0	24.0.00-2384

Display Release Information

Use one of the following commands to display the release of the agent you're running and optional details about added functionality.

- Display release number
- Display release number with details

Display Release Number

To display the release number of the agent you are running, use the following command:

```
cybAgent -v
```

The `cybAgent -v` command displays both the version and build level of the agent you're running. It also lists the enabled plug-ins associated with the agent, as shown in the following examples.

Windows

```
C:\Program Files\CA\WorkloadAutomationAE\SystemAgent\AGENT_ISO>cybAgent.exe -v

Workload Automation Agent for:
   Windows  64-bit
   Version  24.0.00-7794

Enabled plug-in(s):
   - runner
   - router
   - filemon
   - objmon
   - filebrowser
   - ftp

Enabled Advanced Integration(s):

Built at: Sep 11 2024, 00:28:17
```

UNIX/Linux/IBM i/HP NonStop

```
[root@skata-2022-09-13-11-27-33 AGENT_ISO]# ./cybAgent -v

Workload Automation Agent for:
   Linux x86_64 64-bit
   Version  24.0.00-7794

Enabled plug-in(s):
   - runner
   - router
   - filemon
   - objmon
   - filebrowser
   - ftp

Enabled Advanced Integration(s):

Built at: Sep 10 2024, 21:27:37
```

Display Release Number with Details

To display the release number along with details about the release you're running, including any extended functionality installed with the agent, run the following command:

```
cybAgent -vv
```

The `cybAgent -vv` command lists all installed plug-ins, integrations, and advanced integrations. It also provides each component's enablement status and build levels, as shown in the following examples.

Windows

```
C:\Program Files\CA\WorkloadAutomationAE\SystemAgent\AGENT_ISO>cybAgent.exe -vv

Workload Automation Agent for:
   Windows  64-bit
   Version  24.0.00-7794

Functionality Installed with Agent:
   Name                              Enabled
   - config                            No
   - filebrowser                       Yes
   - filemon                           Yes
   - ftp                               Yes
   - management                        No
   - objmon                            Yes
   - router                            Yes
   - runner                            Yes
   - snmp                              No

Additional Integrations/Plugins Installed on Agent:
   Name                              Enabled   Version
   - Application Services              No      24.0.00-3106
   - Database                          No      24.0.00-2333
   - Web/Application Services [Data]   No      24.0.00-3106
   - Informatica                       No      24.0.00-2252
   - Web Services [PAM]                No      24.0.00-3106
   - Micro Focus                       No      24.0.00-2186
   - Oracle E-Business Suite           No      24.0.00-2157
   - Remote Execution                  No      24.0.00-2384
   - PeopleSoft                        No      24.0.00-2616
   - SAP                               No      24.0.00-2642
   - MS SQL Server                     No      24.0.00-2193
   - Web Services                      No      24.0.00-3106
   - K8s                               Yes     1.0.78
   - AWS Lambda                        No      1.0.8
   - AWS StepFunction                  No      1.0.9

Additional Advanced Integrations Installed on Agent: None
Plugin Extensions Configured on Agent: None
Built at: Sep 11 2024, 00:28:17
```

UNIX/Linux/IBM i/HP NonStop

```
[root@skata-2022-09-13-11-27-33 AGENT_ISO]# ./cybAgent -vv
Workload Automation Agent for:
   Linux x86_64 64-bit
   Version  24.0.00-7794

Functionality Installed with Agent:
   Name                              Enabled
   - config                            No
   - filebrowser                       Yes
   - filemon                           Yes
```

```
   - ftp                                Yes
   - management                         No
   - objmon                             Yes
   - router                             Yes
   - runner                             Yes
   - snmp                               No

Additional Integrations/Plugins Installed on Agent:
   Name                                Enabled   Version
   - Database                            No      24.0.00-2333
   - Web Services                        No      24.0.00-3106
   - Web Services [PAM]                  No      24.0.00-3106
   - Web/Application Services [Data]     No      24.0.00-3106
   - Application Services                No      24.0.00-3106
   - MS SQL Server                       No      24.0.00-2193
   - Remote Execution                    No      24.0.00-2384
   - PeopleSoft                          No      24.0.00-2616
   - Oracle E-Business Suite             No      24.0.00-2157
   - SAP                                 No      24.0.00-2642
   - Informatica                         No      24.0.00-2252
   - Micro Focus                         No      24.0.00-2186
   - K8s                                 Yes     1.0.78
   - AWS Lambda                          No      1.0.8
   - AWS StepFunction                    No      1.0.9

Additional Advanced Integrations Installed on Agent:
   Name                                Enabled   Version
   - Hadoop                              No      24.0.00-2644

Plugin Extensions Configured on Agent: None
Built at: Sep 10 2024, 21:27:37
```

Acknowledgments: Third-Party Software Agreements

2

Workload Automation System Agent and Plug-Ins use third-party software in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements.

Download the Workload Automation System Agent 24.0 TPSR's to read the license agreements.

Getting Started

This documentation is designed for system administrators who are getting started with Workload Automation Agents. The information is targeted to individuals 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.

NOTE

- 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.

Overview of WA Agent for Remote Execution

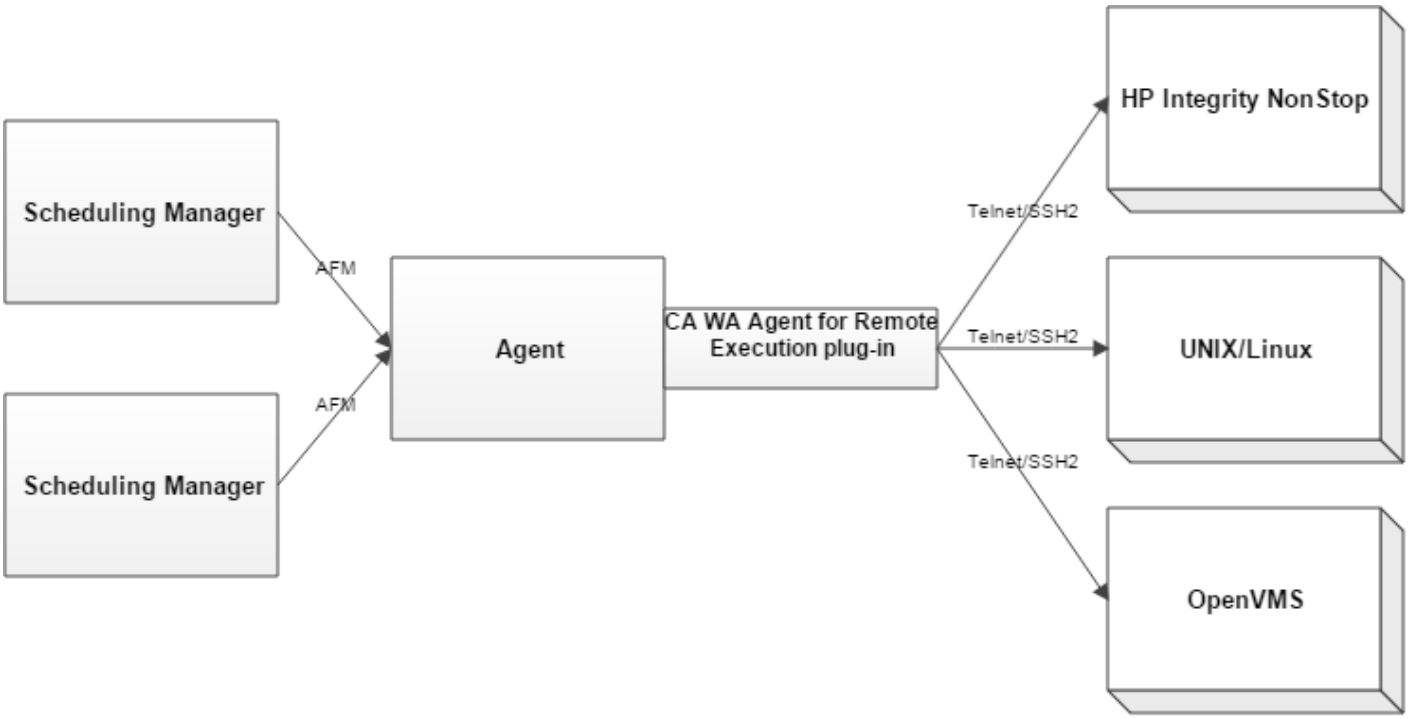
WA Agent for Remote Execution lets you connect to remote systems on UNIX, HP Integrity NonStop (Tandem), Windows, and OpenVMS. You can connect to the remote system using Telnet or Secure Shell (SSH2). With the WA Agent for Remote Execution, you can define and run remote execution jobs.

The 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 WA Agent for Remote Execution plug-in, and the remote system:

Figure 1: RemoteExecutionAgent2



Job Types Supported by WA Agent for Remote Execution

With the WA Agent for Remote Execution, you can define and run a Remote Execution job. This job type 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 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.

NOTE

For detailed information about agent architecture and the way that agents communicate with schedulers, refer to the Getting Started section in the System Agent documentation.

Installing or Upgrading

To install the agent plug-in, review the types of information you will need to provide and the decisions you need to make in advance. Then proceed with the installation.

> **NOTE**
>
> For information about installing the System Agent, see Installing or Upgrading in the Workload Automation Agent documentation.

Prepare for Installation

Review the System Requirements

To ensure that your system meets the minimum requirements for installation, review the system requirements.

Decide 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 ESP dSeries Workload Automation, 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 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.

Download the Agent and Agent Plug-Ins

You can download the Workload Automation System Agent and associated agent plug-ins from the Broadcom Support website. Use one of the following procedures, based on the Workload Automation scheduler you use.

> **NOTE**
> All the plugin .cfg files are added to the addons directory in the agent install directory upon System Agent install.

ESP dSeries Workload Automation/Workload Automation DE Users

1. Log into Broadcom Download Center.
2. In the **Search You Product** field search for **workload automation**.
3. Select from the list of shown products.
4. Select **MULTI-PLATFORM** along with the release and service pack you want from the **RELEASE** and **SERVICEPACK** drop-down menus.

5. Click either **CART** or **DOWNLOAD** and complete the selected procedure.

AutoSys Workload Automation Users/Workload Automation AE

1. Log into Broadcom Download Center.
2. In the **Search You Product** field search for **workload automation**.
3. Select from the list of shown products.
4. Select **MULTI-PLATFORM** along with the release and service pack you want from the **RELEASE** and **SERVICEPACK** drop-down menus.
5. Click either **CART** or **DOWNLOAD** and complete the selected procedure.

Workload Automation CA 7 Edition Users

1. Log into Broadcom Download Center.
2. In the **Search You Product** field search for **workload automation agents**.
3. Select **WORKLOAD AUTOMATION CA7 EDITION** from the list of shown products.
4. Select **Workload Automation Agent CA7 Edition MVS** along with the release and service pack you want from the **RELEASE** and **SERVICEPACK** drop-down menus.
5. Click either **CART** or **DOWNLOAD** and complete the selected procedure.

Workload Automation ESP Edition Users

1. Log into Broadcom Download Center.
2. In the **Search You Product** field search for **workload automation**.
3. Select **WORKLOAD AUTOMATION ESP EDITION** from the list of shown products.
4. Select **Workload Automation Agent ESP Edition MVS** along with the release and service pack you want from the **RELEASE** and **SERVICEPACK** drop-down menus.
5. Click either **CART** or **DOWNLOAD** and complete the selected procedure.

Prerequisites for Remote Agent

Ensure that the following prerequisites are met.

Remote Agent on Windows Systems

1. Install the OpenSSH server.
   For more information, see the Microsoft Documentation.
2. Grant **RemoteEnable** permissions for WMI. This permission allows remote access to WMI services.
   a. Open **Computer Management** on the target Windows services.
   b. Navigate to **WMI Control** > **Properties**.
   c. Under the **Security** tab, select the namespace.
   d. Click **Security**, then add the user or group and grant the **RemoteEnable** permission.
3. Grant Access to the Spool Location.
   a. The spool directory is typically located in **C:\Windows\System32\spool**.
   b. Right-click the spool folder, select **Properties**.
   c. Under the **Security** tab, grant the user the necessary permissions to access the folder.

Install WA Agent for Remote Execution

The WA Agent for Remote Execution installs into the Workload Automation Agent installation directory.

Installation

To install Workload Automation Agent for Remote Execution, follow these steps:

1. Install Workload Automation System Agent.
   For more information, see Installing and Upgrading in the System Agent documentation.
2. Apply the latest Workload Automation System Agent agent patch to the platform where you install the agent plug-in.
3. Complete the Prerequisites.
4. Install the agent plug-in using one of these methods:
   - Install using an interactive program.
   - Install using a silent installer.
5. Create a custom properties file for the remote system.
6. (HP-UX remote systems only) Enable UNIX95 behavior.
7. Configure the system properties for the remote system, if necessary.

NOTE
After you install the agent plug-in, configure the scheduling manager to work with the agent plug-in. For more information, use the links for your schedule manager found under Additional Resources.

Install the Agent Plug-in Using an Interactive Program

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

Follow these steps:

1. Change to the agent installation directory. For example, type the following command:
   - On UNIX:
     ```
     cd opt/CA/WA_Agent
     ```
   - On Windows:
     ```
     cd C:\Program Files\CA\WA Agent
     ```
2. Stop the agent using one of the following commands:
   - On UNIX:
     ```
     ./cybAgent -s
     ```
   - On Windows:
     ```
     cybAgent -s
     ```
3. Configure the plug-in.
   - On UNIX:
     ```
     ./PluginInstaller addons/proxy.cfg .
     ```
   - On Windows:
     ```
     PluginInstaller.exe addons\proxy.cfg .
     ```

   You are prompted to enable the default alias.

   NOTE
   The default alias for WA Agent for Remote Execution is *agentname*_PROXY.
4. 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.
5. Start the agent using one of the following commands:
   - On UNIX:
     ```
     ./cybAgent -a
     ```
   - 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 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.

Follow these steps:

1. Change to the agent installation directory. For example, type the following command:
   - On UNIX:
     ```
     cd opt/CA/WA_Agent_R11
     ```
   - On Windows:
     ```
     cd C:\Program Files\CA\WA Agent
     ```
2. Stop the agent using one of the following commands:
   - On UNIX:
     ```
     ./cybAgent -s
     ```
   - On Windows:
     ```
     cybAgent -s
     ```
3. Run the silent installer using the following command:
   ```
   PluginInstaller proxy.cfg . 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:

   ```
   ./PluginInstaller addons/proxy.cfg . . path/silent.txt false
   ```

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

Follow these steps:

1. Open the silent.txt file using a text editor.

   NOTE
   The letter Y on the first line of the file corresponds to the prompt asking whether Workload Automation Agent R11.3, Service Pack 4, Incremental 2 or higher is installed. The letter N on the second 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 third 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:

```
Y
N
A
```

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

```
[root@lvndev002548 WA_AGENT1]# ./PluginInstaller addons/proxy.cfg . ./silent.txt  false

This version of Workload Automation Agent for Remote Execution requires
Workload Automation Agent 12.0.00 Build 6000 or later.
To verify the current version of Workload Automation Agent, run the following command: cybAgent -v

Is Workload Automation Agent 12.0.00 Build 6000 or higher installed? (Y/N): Y
Do you want to set up the alias? (Y/N): n
Updating agent configuration file /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/agentparm.txt ...
Agent configuration file has been updated
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/unixssh2.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/unixtelnet.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/tandemssh2.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/tandemtelnet.properties ...
```

```
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/openvmsssh2.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/openvmstelnet.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/CAWA-proxyAgent.license ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/silent.txt ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/sampleUnix.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/sampleTandem.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/sampleOpenvms.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/sampleWin.properties ...
Extracting /opt/CA/WorkloadAutomationAE/SystemAgent/WA_AGENT1/config/proxy/windowsssh2.properties ...
Installer successful
```

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. All of the following properties are required.
   - **host**
     Specifies the IP address or host name of the 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
     - windows
     - unix
     - openvms

     > **NOTE**
     >
     > - 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.

NOTE

- 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.

NOTE

- 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. 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.

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.

```
host=myunixhost
port=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]
```

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

```
host=<host>
type=windows
port=22
protocol=ssh2
spoolHome=c:\\user1\\spool
default.user=<user>
default.password=<encrypted password>
pathSeparator=\\
```

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:

```
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.

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:

  ```
  fileTransfer.template=ftp://ftp.funet.fi:1111$PATH
  ```

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

  ```
  ftp://ftp.funet.fi:1111/tmp/CAWA_spool/STRESS/MAIN/PROXY.1/GAR1.TXT
  ```

  NOTE

  - 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: \r\n
  – OpenVMS: \r\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 (for login shell of sh, ksh or bash): 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 unixtelnet.properties file for a UNIX system that the agent connects to using Telnet.

> **NOTE**
> Telnet requires a three-step login.

```
lineTerminator=\n
loginTask.1.startPrompt=ogin:
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=(([^'])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
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 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:

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

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

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.

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

Example: System Properties for Connecting to a Windows System Using SSH2

```
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=40000
endPromptTimeout=40000
```

Create an Alias

During the agent plug-in installation, you are prompted to create a default alias, which you can change after installation. 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.

If you enable an alias on the agent plug-in, also configure the alias on the scheduling manager.

Follow these steps:

1. 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.
2. 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.

Removing

When you no longer need it, you can remove the plug-in agent 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.
11. (Optional) Remove the agent plug-in from the scheduling manager.

    NOTE
    For detailed instructions to remove the agent from the scheduling manager, see the documentation for your scheduling manager.

Example: Renumber the plugins.start_internal_*n* Parameter

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

```
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 the WA Agent for Remote Execution, you would modify the agentparm.txt file as follows:

```
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:

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

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

```
#communication.alias_1=AGENTNAME_PROXY
communication.alias_1=AGENTNAME_MF
```

Configuring

After installing the agent, you can configure the required behavior, such as communicating with remote systems and automatically cleaning spool files.

- Override System Properties
- Delete Spool Files and Job Logs

Override 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.

The following examples provide details for different scenarios.

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.

Delete Spool Files and Job Logs

The WA Agent for Remote Execution does *not* support the following agent settings to clean job log and spool files. Due to this restriction, you cannot configure the agent to delete automatically the spool files of completed Remote Execution jobs or job logs.

```
oscomponent.joblog.success.autocleanup=true
agent.spool.success.autocleanup=true
```

However, A WA Agent for Remote Execution supports automatic cleanup of spool files through the following agent settings:

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

Reference

The topics in the Reference section provide information about the parameters you can use to configure agent behavior.

- Plug-In Specific Parameters
- Remote Systems Usage Information

Plug-In Specific Parameters

You can configure the following parameters in the agentparm.txt file for the WA Agent for Remote Execution. 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.

Default: *agentname*_PROXY

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

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 *hostname*command.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.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)

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 Broadcom.

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 Broadcom.

proxy.ssh2.sshd.connect.timeout

Specifies the connection timeout value in milliseconds (ms) for the TCP connection using SSH2.

Default: 30000 (ms)

proxy.ssh2.sshd.timeout

Controls the read timeout, in milliseconds (ms), for the TCP connection using SSH2.

Default: 30000 (ms)

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 Broadcom.

> **NOTE**
>
> For information about System Agent parameters, see WA Agent for UNIX, Linux, Windows, or IBM i Agent Parameters.

Remote Systems Usage Information

The agent can connect to up to six remote systems. For example, you can connect the agent to four remote systems on UNIX and two remote systems on OpenVMS. If you try to connect to more than six remote systems, the job will end with a submission error.

The following sections provide usage information specific to UNIX, HPE Integrity NonStop, and OpenVMS remote systems.

> **NOTE**
> For a list of supported platforms by version, see Remote Platforms Supported by the Plug-In in the Workload Automation Agent for Remote Execution documentation.

UNIX Remote Systems

Environment Variables

On UNIX systems, WA Agent for Remote Execution sets the value of an environment variable but does not provide additional handling. WA Agent for UNIX has some special handling for PWD, STDOUT, and some other variables. For example, setting PWD using the WA Agent for UNIX will result in switching a working directory to the specified location. Setting PWD using the WA Agent for Remote Execution will set the PWD but will not change the working directory. This behavior is consistent with what happens when setting PWD using the command line.

Special Characters

The following special characters are not supported for use in executing commands on UNIX systems: double quote (“), single quote (‘), and semicolon (;).

Prompts in Job Spool Files Are Not Supported

The agent truncates the spool file output when it contains the $CA_PROMPT string resulting in an incomplete file. When the agent reaches a string containing a prompt, it treats the string as the end of the response.

HPE Integrity NonStop Remote Systems

Supported Completion Codes

WA Agent for Remote Execution only supports completion codes 1 through 6. For other completion codes returned by the system, use the job spool file to confirm expected behavior, for example, whether a file is deleted.

Migrating the Agent Parameter File from a Legacy Telnet Agent

The WA Agent for Remote Execution replaces the legacy Telnet agent for Tandem. If you run Tandem jobs on the legacy agent, you can manually migrate your version 2 agent parameter file to the new agent.

The following table maps the parameters from your version 2 agent parameter file to the new agent. Except for the following parameters, you can ignore the other parameters from the legacy Telnet agent.

V2 Agent Parameter File	WA Agent
CryptKey	security.cryptkey in agentparm.txt To use the same encryption method and key, choose DES as the encryption method during the agent installation. You can also choose a different method such as AES or use a different key. **Note:** Use the same encryption method and key on the scheduling manager.
WaitFirstSysPrompt	login.endPromptTimeout in either the custom or system properties file
AgentName	Agentname in agentparm.txt file Specify during the agent installation when prompted for the agent name.
RequestPort	communication.inputport in agentparm.txt file Specify during the agent installation when prompted for the input port.
ManagerAddress	communication.manageraddress_1 in agentparm.txt file Specify during the agent installation when prompted for the manager address.
ManagerPort	communication.managerport_1 in agentparm.txt file Specify during the agent installation when prompted for the manager port.
MonitorObject	communication.monitorobject_1 in agentparm.txt file
CompletionCode	completionCode in either the custom or system properties file

OpenVMS Remote Systems

Using a Job Queue Other than SYS$BATCH

If the job is not using the default job queue SYS$BATCH, you must specify the job queue in the job definition. For example, to use SYS$BATCH_I64SYS as the job queue, specify the following submit modifier in the job definition:

```
SUBMIT_MODIFIER /QUEUE=SYS$BATCH_I64SYS
```

NOTE
For more information about specifying submit modifiers, see your scheduling manager documentation.

Specifying the Maximum Number of Concurrent Jobs

You can specify the maximum number of concurrent jobs that can run on a job queue by setting the JOB_LIMIT attribute. For example, to run up to 50 jobs concurrently on the SYS$BATCH job queue, issue the following command on the OpenVMS system:

```
SET QUEUE SYS$BATCH /JOB_LIMIT=50
```

When the maximum number of concurrent jobs specified by JOB_LIMIT is reached, subsequent job submissions are queued until jobs complete running. As a result, the actual time a job runs can be different than the time the job is submitted.

Using Unsupported Characters to Identify the Job

If the spool file location is not specified, the agent uses the job ID to create the directory that stores temporary files and spool files. The job ID identifies the job on the scheduling manager and is composed of the job name and other values. To comply with the OpenVMS directory and file specification, we recommend that you identify the job using only alphanumeric characters, hyphen, dollar sign, and underscore.

Migrating the Agent Parameter File from a Legacy Telnet Agent

The WA Agent for Remote Execution replaces the legacy Telnet agent for OpenVMS. If you run OpenVMS jobs on the legacy agent, you can manually migrate your version 2 agent parameter file to the new agent.

The following table maps the parameters from your version 2 agent parameter file to the new agent. Except for the following parameters, you can ignore the other parameters from the legacy Telnet agent.

V2 Agent Parameter File	WA Agent
CryptKey	security.cryptkey in agentparm.txt To use the same encryption method and key, choose DES as the encryption method during the agent installation. You can also choose a different method such as AES or use a different key. **Note:** Use the same encryption method and key on the scheduling manager.
WaitFirstSysPrompt	login.endPromptTimeout in either the custom or system properties file
AgentName	Agentname in agentparm.txt file Specify during the agent installation when prompted for the agent name.
RequestPort	communication.inputport in agentparm.txt file Specify during the agent installation when prompted for the input port.
ManagerAddress	communication.manageraddress_1 in agentparm.txt file Specify during the agent installation when prompted for the manager address.
ManagerPort	communication.managerport_1 in agentparm.txt file Specify during the agent installation when prompted for the manager port.
MonitorObject	communication.monitorobject_1 in agentparm.txt file

Migrating Job Definitions from a Legacy Telnet Agent

If you run OpenVMS jobs on the legacy agent, you must update your old job definitions to run them on the new agent.

The legacy Telnet agent provided a command procedure named ESPSUBMIT.com, which is not required by the new agent. To migrate your old job definitions to the new agent, do the following:

- Replace ESPSUBMIT.COM with the actual commands you intend to run. For example, assume that your existing job definition uses ESPSUBMIT.COM as the command and PIC_TIME.COM as the argument. Your new job definition would use PIC_TIME.COM as the command without an argument.
- If your existing job definition specifies any options to pass to ESPSUBMIT.COM, such as /USER, specify these values as described in your scheduling manager documentation.
- Instead of using /PARAMETERS to specify arguments, include the arguments as part of the command string. For example, assume that your existing job definition uses ESPSUBMIT.COM as the command and "PIC_TIMER.COM /

PARAMETERS="RECORD"" as the argument. Your new definition would use "@PIC_TIMER.COM "RECORD"" as the command without an argument.

- Instead of using /LOG_FILE, specify the path to the spool file as described in your scheduling manager documentation.

Troubleshooting

The articles in this section can help you perform basic troubleshooting procedures for Workload Automation Agent for Remote Execution. For more information about troubleshooting the System Agent, see Troubleshooting in the Workload Automation Agent documentation.

During Service Request investigations, Product Support Services commonly requires log files to help resolve your problem.

Job Goes in State SUBERROR Failed Due to Submission Problem

Valid on all remote systems

Symptom:

A Remote Execution job goes in state SUBERROR Failed and produces an error message like the following message:

```
State SUBERROR Failed Status(Submission problem) LStatus("Logon failed due to timeout. Last response is ..You
 entered an invalid login name or password....
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.

Valid on UNIX and Linux systems

Symptom:

A Remote Execution job goes in state SUBERROR Failed and produces an error message like the following message:

```
State SUBERROR Failed Status(Submission problem) LStatus("java.lang.Exception: execute
 command stty -echo echonl failed due to timeout. Last response is %")
```

Solution:

The message can occur when the login shell for the user is csh or tcsh, but the connection properties files have default settings for sh, ksh, or bash.

To set the connection properties for csh and tcsh, add the following line to the <*target_machine*>.properties file:

```
promptTask.1.command=set prompt='CA'_'PROMPT '
```

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.

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:

```
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:

```
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.

For troubleshooting topics related to the system agent, see Troubleshooting in the System Agent documentation.

Remote execution plugin

Issue/Introduction

Set up a test job for Remote Execution and it fails with "Logon failed due to timeout."

autorep -j test_remote_execution -d

```
Job Name              Last Start               Last End         ST/Ex Run/Ntry Pri/Xit
_____________________________________________________________________________
test_remote_execution         03/30/2022 12:40:44  03/30/2022 12:40:44  FA    310800492/1 1
 Status/[Event]  Time                 Ntry ES  ProcessTime           Machine
--------------  -------------------- --  --  -------------------- ----------------------------------------
 STARTING        03/30/2022 12:35:17    1  PD  03/30/2022 12:35:18   <machine>
 RUNNING         03/30/2022 12:40:44    1  PD  03/30/2022 12:40:45   <machine>
 FAILURE         03/30/2022 12:40:44    1  PD  03/30/2022 12:40:45


<Logon failed due to timeout. Last response is ast login: Wed Mar 30 12:38:52 2022 from 10.68.40.240...Autosys
 Required Environment Variable Settings..****** <target server host> is not a supported AutoSys Scheduler/
Application server..(user@host>


 [*** ALARM ***]
 JOBFAILURE    03/30/2022 12:40:45    1  PD  03/30/2022 12:40:47   qautoad
```

Cause

```
Logon timeout:03/30/2022 12:35:17.908-0400 5 ProxyPlugin.proxy Internal
 Thread.CybWOBRunProxyHandler.processWob[:86] - Processing WOB 788194.310800492_1/WAAE_WF0.1/MAIN:
 RemoteTarget(hostname) SpoolFile(/nfs/tmp/ken/remote_execution.out) ExitCode(0,S) Command1(df -h /export)
 User(...) Password(...) MFUser(...) WOBRequestID(312BA1821FB4B94E450F3AB43FBDCDD25014F37316486581179020)
```

```
03/30/2022 12:35:18.381-0400 5 ProxyPlugin.proxy Internal Thread.CybWOBRunProxyHandler.processWob[:145] -
 Submission thread started for WOB : 788194.310800492_1/WAAE_WF0.1/MAIN: RemoteTarget(hostname) SpoolFile(/
nfs/tmp/ken/remote_execution.out) ExitCode(0,S) Command1(df -h /export) User(...) Password(...) MFUser(...)
 WOBRequestID(312BA1821FB4B94E450F3AB43FBDCDD25014F37316486581179020)
03/30/2022 12:35:18.381-0400 5 ProxyPlugin.788194.310800492_1/WAAE_WF0.1/
MAIN[Proxy].ProxyConnection.<init>[:139] - Creating connection to qora16dbadm01 for WOB 788194.310800492_1/
WAAE_WF0.1/MAIN
03/30/2022 12:35:18.381-0400 5 ProxyPlugin.788194.310800492_1/WAAE_WF0.1/
MAIN[Proxy].ProxyConnection.<init>[:159] - Processing destination properties
03/30/2022 12:35:18.382-0400 5 ProxyPlugin.788194.310800492_1/WAAE_WF0.1/
MAIN[Proxy].ProxyConnection.<init>[:175] - Creating ConnectionInfo
03/30/2022 12:35:18.383-0400 5 ProxyPlugin.788194.310800492_1/WAAE_WF0.1/
MAIN[Proxy].ProxyConnection.<init>[:239] - Connecting...
03/30/2022 12:35:59.670-0400 1 ProxyPlugin.788194.310800492_1/WAAE_WF0.1/
MAIN[Proxy].ProxyConnection.connect[:354] - Exception: java.lang.Exception: Logon failed due to timeout. Last
 response is ast login: Wed Mar 30 12:18:41 2022 from 10.68.40.240

Autosys Required Environment Variable Settings
****** remotehost is not a supported AutoSys Scheduler/Application server
(user@host)#  ;wobId = 788194.310800492_1/WAAE_WF0.1/MAIN ;retryCount = 0
.......

03/30/2022 12:40:44.968-0400 1 ProxyPlugin.788194.310800492_1/WAAE_WF0.1/
MAIN[Proxy].ProxyConnection.<init>[:246] - java.lang.Exception: Logon failed due to timeout. Last response is
 ast login: Wed Mar 30 12:38:52 2022 from 10.68.40.240

Autosys Required Environment Variable Settings
****** remotehost is not a supported AutoSys Scheduler/Application server
(user@host)#
at cybermation.plugins.proxy.tasks.SSH2Connection.connectAndLogon(SSH2Connection.java:233)
at cybermation.plugins.proxy.ProxyConnection.connect(ProxyConnection.java:323)
at cybermation.plugins.proxy.ProxyConnection.<init>(ProxyConnection.java:240)
at cybermation.plugins.proxy.RemoteExecutionHandler.run(RemoteExecutionHandler.java:217)
at java.lang.Thread.run(Thread.java:821)
03/30/2022 12:40:44.968-0400 5 ProxyPlugin.788194.310800492_1/WAAE_WF0.1/
MAIN[Proxy].ProxyConnection.disconnect[:372] - Disconnecting connection for 788194.310800492_1/WAAE_WF0.1/MAIN
03/30/2022 12:40:44.968-0400 5 ProxyPlugin.788194.310800492_1/WAAE_WF0.1/
MAIN[Proxy].ProxyConnection.disconnect[:375] - Finished disconnecting for 788194.310800492_1/WAAE_WF0.1/MAIN

Environment:CA Workload Automation AE 12.0.01
Workload Automation Agent 12.0, Build 6181, Service Pack 00, Maintenance Level 00
```

Resolution

Raise the value for login.endPromptTimeout to 120000 in unixssh2.properties and restart the agent.

Additional Resources

This topic provides additional information, including product support, documentation, and other resources for the agent.

Broadcom Links

- Broadcom Enterprise Automation
- Product Support
- Communities

Workload Automation Agent Documentation

- Workload Automation System Agent
- Workload Automation Agent for Application Services
- Workload Automation Agent for Databases
- Workload Automation Agent for Informatica
- Workload Automation Agent for Kubernetes
- Workload Automation Agent for Micro Focus
- Workload Automation Agent for Microsoft SQL Server
- Workload Automation Agent for Oracle E-Business Suite
- Workload Automation Agent for PeopleSoft
- Workload Automation Agent for Remote Execution
- Workload Automation Agent for SAP
- Workload Automation Agent for Web Services

Workload Automation Scheduler Documentation

- AutoSys Workload Automation
- Workload Automation CA 7 Edition
- ESP dSeries Workload Automation
- Workload Automation ESP Edition

Workload Automation Integration Documentation

- Workload Automation Advanced Integration for Hadoop
- Workload Automation Advanced Integration for SAP Business Warehouse
- Workload Automation Advanced Integration for SAP Solution Manager

Workload Automation Agent Add-on Documentation

- Workload Automation Agent Monitor

Workload Automation Mainframe Agent Documentation

- Workload Automation for z/OS

Archived Documentation

This page describes provides access to PDF/text files for Workload Automation Agents bookshelf and legacy documentation (two forms documentation for previous releases). For up to date Workload Automation Agent Documentation for supported versions, see Additional Resources.

Product	Release	Documentation Link
Agent for UNIX, Linux, Windows, or i5/OS		
	11.3.3	• Agent for UNIX, Linux, or Windows Implementation Guide • Agent for i5/OS Implementation Guide • Agent for UNIX, Linux, Windows, and i5/OS Release Notes
	11.3.2 Cumulative 1	• Release Notes
	11.3.1 Cumulative 4	• Release Notes
	11.3	• Agent for UNIX, Linux, or Windows Implementation Guide • Agent for i5/OS Implementation Guide • Automation Agent for UNIX, Linux, or Windows Readme • Agent for i5/OS Readme • Agent for UNIX, Linux, or Windows Release Notes • Agent for i5/OS Release Notes
Agent for i5/OS		
	11.3	• Agent for i5/OS Implementation Guide • Agent for i5/OS Readme • Agent for i5/OS Release Notes
Workload Automation System Agent (for UNIX, Linux, or Windows)		
	11.3.3	• Release Notes
	11.3.2	• Release Notes
	11.3.1	• Release Notes
Agent for Application Services		
	11.3.3	• Implementation Guide • Release Notes
	11.3.1	• Release Notes
	11.3	• Implementation Guide • Readme • Release Notes
Agent for Databases		
	11.3.2	• Implementation Guide • Release Notes

	11.3	• Implementation Guide • Readme • Release Notes
	1.1	• ESP Agent for Databases Release 1.1: Guide to Scheduling Workload
	1.0	• Cybermation ESP Agent for Databases Release 1.0: Installation and Setup Guide
Agent for HPE Integrity NonStop *(see top row of table for later version)*		
	11.3.1	• Implementation Guide • Release Notes
	2.0	• ESP Agent for the NSK Operating System Version 2.0 Installation Guide • ESP Agent for the NSK Operating System Version 2.0 User's Guide
Agent for Informatica		
	11.3.3	• Release Notes
	11.3.1	• Implementation Guide • Release Notes • CLI User's Guide
Agent for Micro Focus		
	11.3.3	• Implementation Guide • Release Notes
	11.3.2	• Release Notes
	11.3	• Implementation Guide • Readme • Release Notes
	1.0	• Cybermation ESP Business Agent for Micro Focus Enterprise Server: Installation and Setup Guide
Agent for Microsoft SQL Server		
	11.3.3	• Implementation Guide • Release Notes
	11.3.1	• Implementation Guide • Release Notes • CLI User's Guide
Agent for Oracle E-Business Suite		
	11.3.3	• Implementation Guide • Release Notes
	11.3.2	• Implementation Guide • Release Notes
	11.3.1	• Implementation Guide • Readme • Release Notes

	11.3	• Implementation Guide • Readme • Release Notes
	3.1	• Cybermation ESP Business Agent for Oracle E-Business Suite Release 3: Administrator's Guide • ESP Business Agent for Oracle E-Business Suite Release 3: Guide to Scheduling Workload • ESP Business Agent for Oracle E-Business Suite Release 3.1: Installation and Setup Guide • ESP Business Agent for Oracle E-Business Suite Release 3.1: Release Notes
	3.0	• ESP Business Agent for Oracle E-Business Suite Release 3: Administrator's Guide • ESP Business Agent for Oracle E-Business Suite Release 3: Guide to Scheduling Workload • Cybermation ESP Business Agent for Oracle E-Business Suite Release 3: Release Notes
Agent for PeopleSoft		
	11.3.3	• Release Notes
	11.3.2	• Release Notes
	11.3	• Implementation Guide • Readme • Release Notes
	2.1	• ESP Business Agent for PeopleSoft, UNIX Edition, Release 2 , Service Pack 1: Guide to Scheduling Workload • ESP Business Agent for PeopleSoft® Release 2, Service Pack 1 Windows® Edition: Guide to Scheduling Workload • ESP Business Agent for PeopleSoft Release 2.1: Installation and Setup Guide • ESP Business Agent for PeopleSoft Release 2.1: Release Notes
	2.0	• Cybermation ESP Business Agent for PeopleSoft® Release 2, Service Pack 4: Release Notes
Agent for Remote Execution		
	11.3.3	• Release Notes
	11.3.1	• Implementation Guide • Readme • Release Notes
Agent for SAP		

	11.3.3	• Release Notes
	11.3.2	• Release Notes
	11.3	• Implementation Guide • Readme • Release Notes
	5.1.1	• Workload Automation Business Agent for SAP Release 5.1, Service Pack 1: Installation and Setup Guide • Workload Automation Business Agent for SAP Release 5.1, Service Pack 1: Release Notes
	5.1	• ESP Business Agent for SAP Solutions Release 5: Guide to Scheduling Workload • ESP Business Agent for SAP Solutions Release 5.1: Installation and Setup Guide • ESP Business Agent for SAP Solutions Release 5.1: Release Notes
	5.0.3	• ESP Business Agent for SAP Solutions Release 5: Admininstrator's Guide • ESP Business Agent for SAP® Solutions Release 5: Guide to Scheduling Workload • Cybermation ESP Business Agent for SAP Solutions Release 5, Service Pack 3: Release Notes
Agent for Web Services		
	11.3.3	• Release Notes
	11.3.1	• Implementation Guide • Release Notes • CLI User's Guide
	11.3	• Implementation Guide • Readme • Release Notes
	11.1	• Implementation Guide • Release Notes

Workload Automation Agents Documentation Home

For Workload Automation Agents documentation, see Workload Automation Agents.

To explore integrations that enhance control, visibility, service delivery, and cloud adoption success, see the Automation Marketplace.

Documentation Legal Notice

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 Broadcom at any time. This Documentation is proprietary information of Broadcom and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Broadcom.

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 Broadcom 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 Broadcom that all copies and partial copies of the Documentation have been returned to Broadcom or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM 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 BROADCOM 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 BROADCOM 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 Broadcom Inc.

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.


BROADCOM®
SOFTWARE