CA Workload Automation Agent for UNIX, Linux, or Windows

Implementation Guide

Release 11.3, Second Edition

technologies

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 may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, 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.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Workload Automation AE
- CA Workload Automation DE
- CA Workload Automation EE
- CA Workload Automation SE
- CA Workload Automation Desktop Client (CA WA Desktop Client)
- CA Workload Automation High Availability DE (CA WA High Availability)
- CA Workload Automation Web Services (CA WA Web Services)
- CA Workload Automation Agent for UNIX (CA WA Agent for UNIX)
- CA Workload Automation Agent for Linux (CA WA Agent for Linux)
- CA Workload Automation Agent for Windows (CA WA Agent for Windows)
- CA Workload Automation Agent for i5/OS (CA WA Agent for i5/OS)
- CA Workload Automation Agent for z/OS (CA WA Agent for z/OS)
- CA Workload Automation Agent for Application Services (CA WA Agent for Application Services)
- CA Workload Automation Agent for Web Services (CA WA Agent for Web Services)
- CA Workload Automation Agent for Micro Focus (CA WA Agent for Micro Focus)
- CA Workload Automation Agent for Databases (CA WA Agent for Databases)
- CA Workload Automation Agent for SAP (CA WA Agent for SAP)
- CA Workload Automation Agent for PeopleSoft (CA WA Agent for PeopleSoft)
- CA Workload Automation Agent for Oracle E-Business Suite (CA WA Agent for Oracle E-Business Suite)
- CA Workload Automation Restart Option EE (CA WA Restart Option)
- CA Spectrum® Service Assurance (CA Spectrum SA)

Contact CA Technologies

Contact CA Support

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

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

Provide Feedback

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

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

Contents

Chapter 1: Introduction 11

Chapter 2: Implementation Checklist 19

Chapter 3: Installing the Agent 21

Chapter 4: Controlling the Agent 47

Chapter 5: Configuring the Agent 55

Chapter 1: Introduction

This section contains the following topics:

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.

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

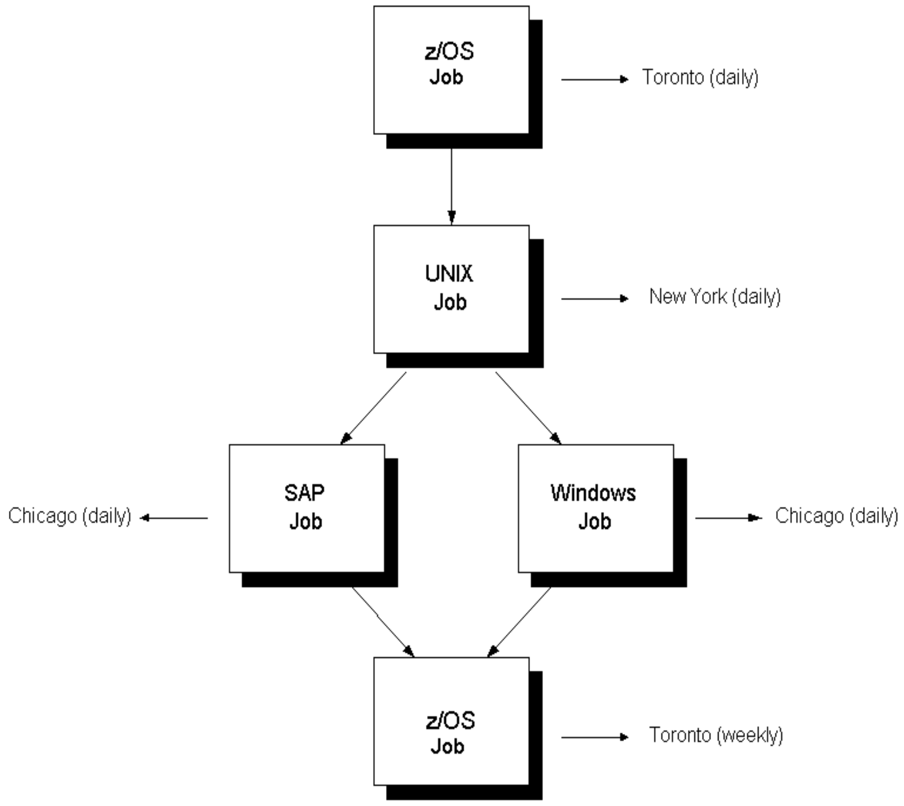
Agents are the key integration components of CA workload automation products. Agents let you automate, monitor, and manage workload on all major platforms, applications, and databases. To run workload on a particular system, you install an agent on that system. If your workload must run on a UNIX computer, for example, you can install and configure the CA WA Agent for UNIX. The agent lets you 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, for example, you can install a database agent plug-in to query and monitor the database. Other agent plug-ins are also available. For more information, 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:



CA WA Agent for UNIX or CA WA Agent for Linux

The CA WA Agent for UNIX or the CA WA Agent for Linux lets a user perform tasks such as the following:

- Run UNIX scripts and execute UNIX commands.
- Monitor file activity and release jobs based on that activity.
- Transfer files using FTP.
- Monitor the agent computer for CPU usage, disk space, IP address, process execution, and text files.
- Retrieve or set the value of an SNMP variable.
- Subscribe for SNMP trap information or publish.

Job Types Supported by CA WA Agent for UNIX or CA WA Agent for Linux

The CA WA Agent for UNIX or the CA WA Agent for Linux lets you define and run the following types of jobs:

CPU Monitoring

Lets you monitor CPU usage.

Disk Monitoring

Lets you monitor disk space.

File Trigger

Lets you monitor file activity and perform an action based on that activity.

FTP

Lets you transfer files using FTP.

IP Monitoring

Lets you monitor an IP address.

Process Monitoring

Lets you monitor process execution.

Secure Copy Protocol

Lets you securely transfer binary files using the Secure Copy Protocol (SCP).

Secure File Transfer Protocol

Lets you securely transfer binary files using the Secure File Transfer Protocol (SFTP).

SNMP Subscribe

Lets you subscribe for SNMP trap information.

SNMP Trap Send

Lets you send SNMP trap information.

SNMP Value Get

Lets you retrieve the value of an SNMP variable.

SNMP Value Set

Lets you set the value of an SNMP variable.

Text File Reading and Monitoring

Lets you search a text file for a string.

UNIX

Lets you run UNIX scripts or execute commands.

Wake on LAN (WOL)

Lets you wake up a computer remotely.

CA WA Agent for Windows

The CA WA Agent for Windows lets a user perform tasks such as the following:

- Run Windows command files.
- Monitor file activity and release jobs based on that activity.
- Transfer files using FTP.
- Monitor the agent computer for CPU usage, disk space, IP address, process execution, and text files.
- Monitor the Windows agent computer for Windows event logs and the status of Windows services.
- Retrieve or set the value of an SNMP variable.
- Subscribe for SNMP trap information or publish.

Job Types Supported by CA WA Agent for Windows

The CA WA Agent for Windows lets you define and run the following types of jobs:

CPU Monitoring

Lets you monitor CPU usage.

Disk Monitoring

Lets you monitor disk space.

File Trigger

Lets you monitor file activity and perform an action based on that activity.

FTP

Lets you transfer files using FTP.

IP Monitoring

Lets you monitor an IP address.

Process Monitoring

Lets you monitor process execution.

Secure Copy Protocol

Lets you securely transfer binary files using the Secure Copy Protocol (SCP).

Secure File Transfer Protocol

Lets you securely transfer binary files using the Secure File Transfer Protocol (SFTP).

SNMP Subscribe

Lets you subscribe for SNMP trap information.

SNMP Trap Send

Lets you send SNMP trap information.

SNMP Value Get

Lets you retrieve the value of an SNMP variable.

SNMP Value Set

Lets you set the value of an SNMP variable.

Text File Reading and Monitoring

Lets you search a text file for a string.

Wake on LAN (WOL)

Lets you wake up a computer remotely.

Windows

Lets you run Windows command files.

Windows Event Log Monitoring

Lets you monitor a Windows event log.

Windows Service Monitoring

Lets you monitor the status of Windows services.

How a Scheduling Manager and Agents Communicate

Agents receive and respond to commands sent by the scheduling manager and transmit data and messages back to the scheduling manager. A scheduling manager and agents communicate by sending Automated Framework Messages (AFMs) to each other. Communication is asynchronous using message queues through TCP/IP ports. For example, while the scheduling manager is sending a new job request to an agent, that agent can be sending completion status for another job.

The following table summarizes the relationship between the scheduling manager and agents:

Scheduling Manager	Agent
Is aware of the entire network	Is aware of the local environment
Sends commands and parameters to the agents	Responds to commands and parameters sent by the scheduling manager
Receives data from the agents	Transmits data to the scheduling manager
Makes decisions	Takes direction from the scheduling manager
Schedules jobs	Runs jobs on different platforms

Receiver Ports

A scheduling manager and agents each have TCP/IP ports to receive messages. The receiver listens on its designated port for messages from one or more senders. When the sender has messages to transmit, it connects to the port of the receiver, sends the messages, and closes the connection.

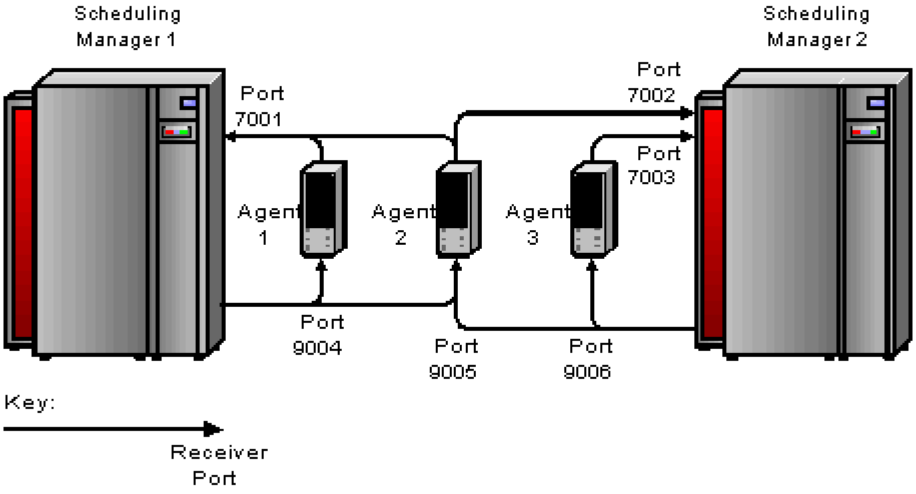
Receiver port configuration is restricted as follows:

- The scheduling manager can have multiple receiver ports (for example, to separate encrypted and unencrypted message traffic). Each of these ports can receive messages from multiple agents.
- An agent has only one receiver port. This port can receive messages from multiple scheduling managers.

Communication Configuration Example

The following diagram shows some possible communication configurations between two scheduling managers and agents.

- Scheduling Manager 1 communicates with Agent 1 and Agent 2 and receives messages from both agents through port 7001. Scheduling Manager 1 sends messages to port 9004 on Agent 1 and port 9005 on Agent 2.
- Scheduling Manager 2 communicates with Agent 2 and Agent 3 and receives messages from Agent 2 through port 7002 and from Agent 3 through port 7003. Scheduling Manager 2 sends messages to port 9005 on Agent 2 and port 9006 on Agent 3.



Chapter 2: Implementation Checklist

This section contains the following topics:

How to Install and Configure the Agent

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

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

To install and configure the agent, follow these steps:

1. Review the system requirements in the *CA Workload Automation Agent for UNIX, Linux, or Windows Release Notes*.
2. Collect information about the scheduling manager (see page 20).
3. Review the agent installation program options (see page 22).
4. (AIX and z/Linux systems only)
   - Install the JRE (see page 29).
   - Set the PATH environment variable (see page 29).
5. Install the agent using one of these methods:
   - Install the agent on UNIX using an interactive program (see page 30) or install the agent on Windows using an interactive program (see page 31).
   - Install the agent using a silent installer (see page 32).

6. Configure the scheduling manager to work with the agent:

   - Define the agent on the scheduling manager.
   - (Optional) Define a user on the scheduling manager.
   - Configure security profiles on the scheduling manager.
   - Verify that the agent works with the scheduling manager.

   Note: For detailed instructions to complete these steps, see the documentation for your scheduling manager (see page 177).

7. (Optional) Configure the agent (see page 55).

8. Configure security features (see page 105).

Collecting Information about the Scheduling Manager

Note: This topic does not apply to CA Workload Automation AE installations.

During the agent installation, you are prompted for information about your scheduling manager. Speak to your administrator and collect the following information:

- Scheduling Manager Name—Corresponds to the Scheduling Manager ID required in the agent installation program
- IP address—Corresponds to the scheduling manager Address required in the agent installation program
- Port number—Corresponds to the Scheduling Manager Port required in the agent installation program

Chapter 3: Installing the Agent

Note: The UNIX instructions in this document also apply to Linux systems unless otherwise noted.

This section contains the following topics:

Agent Installation Considerations for CA Workload Automation AE

CA Workload Automation AE provides its own agent UNIX installation scripts (agent_setup.sh or wa_setup.sh) and Windows installation file (setup.exe). The UNIX scripts and Windows file install and configure the agent specifically for communication with CA Workload Automation AE. For agent installation instructions, refer to the *CA Workload Automation AE UNIX Implementation Guide* or *CA Workload Automation AE Windows Guide*.

This guide covers the agent installation procedures using the agent setup.bin and setup.exe files. Although not recommended, you can use the agent setup.bin and setup.exe files to install the agent to work with CA Workload Automation AE.

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

Installing Multiple Agents on a Single Computer

You can install multiple agents on a single computer. This configuration lets you do the following:

- Distribute the load of the jobs across multiple agents. For example, you can run different jobs for different business applications on the same computer. To run this workload, you can install an agent for one business application and an agent for the other business application and provide access at the agent level.
- Test maintenance applied to an agent before applying maintenance to the production agent.

Important! If a computer with multiple agents is not available, all workload scheduled on that computer is impacted. To avoid a single point of failure, we recommend that you install agents across multiple computers.

Agent Installation Options

The interactive agent installation program prompts you for the following information:

Installation Path

Specifies the path to the location where you want to install the agent program files. The specified location must be empty.

Default:

- For Linux:

  ```
  /CA/WA_Agent_R11_3
  ```

- For UNIX:

  ```
  /CA/WA_Agent_R11_3
  ```

- For Windows:

  ```
  C:\Program Files\CA\WA Agent R11.3
  ```

AgentParm File Conversion

Indicates whether the installation program preserves settings for a Release 6 or Release 7 agent by converting the existing agentparm.txt file.

- Yes—Preserves the parameter settings from the previous release of the agent.
- No—Does not preserve the parameter settings from the previous release of the agent.

Default: No

Path to agentparm.txt file

Specifies the path to an existing agentparm.txt file for a Release 6 or Release 7 agent. When you specify this path, the installation program converts the agentparm.txt file in this directory to an R11.3 version.

Default:

- For UNIX:

  ```
  /Cybermation/ESP System Agent
  ```

- For Windows:

  ```
  C:\Program Files\Cybermation\ESP_System_Agent
  ```

Note: You must include agentparm.txt in the path.

Agent Name

Defines the agent name. You need the agent name when you configure the scheduling manager to work with the agent.

Default: AGENT

Limits: Up to 16 characters; the first character must be a letter

Notes:

- Agent names must begin with an alphabetic character and can contain any alphanumeric characters and the special characters @, $, and underscore (_). Because the scheduling manager uses agent names as file names, use standard file-naming conventions for your operating system.
- For CA Workload Automation DE, the agent name must be in uppercase.

Input Port

Specifies the main port number the agent uses to listen for incoming messages from the scheduling manager. You need this port when you configure the scheduling manager to work with the agent.

Default: 7520

Limits: 1024-65534

Note: On UNIX, ports 1–1023 are reserved ports that require root access.

Number of Managers

Specifies the number of scheduling managers you want to configure to work with the agent.

Default: 1

Manager *n* ID

Specifies the name of the scheduling manager instance that the agent works with, where *n* is an integer that corresponds to the scheduling manager being configured.

Default: CENTRAL_MANAGER

Example: MYSERVER

Note: You can configure the agent to work with multiple scheduling managers by adding additional scheduling manager definitions in the agentparm.txt file.

Manager *n* Address

Specifies the address of the scheduling manager that the agent works with, where *n* is an integer that corresponds to the scheduling manager being configured. This value corresponds to the IP address in the connection details for the scheduling manager. You can specify a list of addresses for the scheduling manager.

Example: 172.24.36.107 (IPv4) or 0:0:0:0:0:FFFF:192.168.00.00 (IPv6)

Notes:

- You can specify a DNS name instead of the IP address for the scheduling manager. However, your agent computer must be able to resolve the DNS name at all times. If there is a DNS outage and your agent computer cannot resolve DNS names, the agent cannot communicate with the scheduling manager.
- If the scheduling manager address never changes, enter the DNS name for the scheduling manager in the hosts file for your agent computer. This entry helps ensure that the IP address can be resolved after DNS disruptions.

Manager *n* Port

Specifies the port that the scheduling manager listens on for communication from agents, where *n* is an integer that corresponds to the scheduling manager being configured. This value corresponds to the port number in the connection details for the scheduling manager.

Default: 7507

Limits: 1024-65534

Cipher Algorithm

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

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

Default: DES

Encryption Key

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

- AES—32 hexadecimal character encryption key.

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

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

DESEDE—48 hexadecimal character encryption key

Local Security Option

Specifies whether local security on the agent is enabled or disabled. Local security on the agent controls which scheduling manager user IDs can perform certain actions, for example, which user IDs can issue CONTROL messages to the agent. If you enable local security, define security rules in a security.txt file.

Default: disabled

Management Connector Option

Enables the following management connectors:

SNMP Connector

Lets you use an SNMP Manager to monitor and control the agent. You can connect the agent to any SNMP Manager that supports SNMP v1, v2, and v3. This option requires the SNMP Manager address and User Datagram Protocol (UDP) port.

Default: disabled

JMX Connector

Lets you use a JMX console to monitor and control the agent. You can use any JMX console that implements JSR-160. This option requires the input port number for the JMX console.

Default: disabled

Remote SNMP Manager Trap Listener Host

Specifies the SNMP trap receiver host name.

Default: localhost

Note: This value applies to the SNMP management connector option.

SNMP Agent Port

Specifies the SNMP GET/SET listening port.

Default: 161

Limits: 1-65535

Note: This value applies to the SNMP management connector option.

JMX Communication Port

Specifies the input port number for the JMX console.

Default: 1099

Limits: 1-65534

Note: This value applies to the JMX management connector option.

Enable FTP Plug-in

Enables the FTP plug-in on the agent, which lets you configure FTP client and FTP server options.

Default: disabled (unselected)

FTP Client Information

Specifies whether the agent can act as an FTP client using Regular or Secure Sockets Layer (SSL) encryption for FTP transfers.

Default: Regular Client Transfer

FTP Server Information

Specifies whether the agent can act as an FTP server using regular or Secure Sockets Layer (SSL) encryption for FTP transfers.

Default: Disable FTP Server

FTP Server Port

Specifies the port number for the agent to act as an FTP server.

Default: 21

Limits: 1-65534

FTP User ID

Specifies the FTP user ID required to access the FTP server.

FTP Password

Specifies the password corresponding to the FTP user ID.

Limits: case-sensitive

Verify FTP Password

Confirms the FTP password.

Enable SNMP Job Type

Specifies whether the agent enables SNMP jobs.

Default: disabled (unselected)

SNMP Trap Listener Port

Specifies the agent port listening for trap information.

Default: 162

Limits: 1-65535

Use SSA Socket

Enables a connection to CA Workload Automation AE using the CA Secure Socket Adapter (SSA).

Default: disabled (unselected)

SSA Socket Path

Specifies the path to the *.so file (UNIX) or *.dll file (Windows) for communication with CA Workload Automation AE using the CA Secure Socket Adapter (SSA).

Default:

- For UNIX:

  ```
  /opt/CA/SharedComponents/Csam/SockAdapter/lib
  ```

- For Windows:

  ```
  C:\Program Files\CA\SharedComponents\Csam\SockAdapter\bin
  ```

Note: For UNIX systems, also specify the CA Workload Automation shared directory.

CA Workload Automation Shared Directory

Specifies the path to the shared components directory for CA Workload Automation AE.

Default: /opt/CA/SharedComponents

Note: This path is required for UNIX systems.

Windows Service Name

Specifies the name for an agent installed on Windows that appears in the list of Services. You can control the agent as a Windows Service.

Default: CA Workload Automation Agent 11.3

Windows Service Option

Sets how the agent, installed as a Windows Service, starts whenever the agent computer is restarted:

- Automatic
- Manual

Default: Manual

User Account Considerations for UNIX Installations

We recommend that you use the root account to install and start the agent on UNIX. Using the root account lets you run jobs under different user accounts.

If you start the agent with an account other than root, you cannot run jobs under different user accounts because the agent cannot switch users. If you plan to run the agent under a specific user account instead of root, consider the following:

- Verify that the user account has the permissions to access the required directories and run the commands and scripts located on the agent computer.
- You can run the agent under the user account when the agent is installed under root. However, you can only run jobs that belong to the user account. We recommend that you install the agent using the specific user account to avoid permission problems.

Install the JRE (for AIX and z/Linux)

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

- On AIX—JRE 1.6 SR8 or higher (The supported JRE is 32-bit. The 64-bit JRE is not supported.)
- On z/Linux—JRE 1.6 SR8 or higher (The supported JRE is 31-bit. The 64-bit JRE is not supported.)

Set the PATH Environment Variable (for AIX and z/Linux)

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

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

java_binary_location

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

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

Install the Agent on UNIX Using an Interactive Program

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

Important! If the installer.properties file is present in the same directory as the setup.bin file and you run the file without any arguments, the agent installs using a silent installation (see page 32).

To install the agent on UNIX using an interactive program

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

   ```
   chmod +x
   ```

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

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

   tempdir

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

6. Type the following command to start the installation:

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

   The agent installation program opens.

7. Press Enter.

   The license agreement appears.

8. Type **y** to accept the license agreement.
9. Continue with the installation by entering the required information.

   Notes:

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

10. Review your selections. To return to a previous option, type **back**.
11. Press Enter to exit the installation program.

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

    Notes:

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

More information:

User Account Considerations for UNIX Installations (see page 29)
Display Debugging Information During Agent Installation (see page 146)

Install the Agent on Windows Using an Interactive Program

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

Important! If the installer.properties file is present in the same directory as the setup.bin file and you run the file without any arguments, the agent installs using a silent installation (see page 32).

To install the agent on Windows

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

   The agent installation program opens.
4. Accept the license agreement and click Next.

   The Product Icons and Shortcuts dialog opens.
5. Continue with the installation by entering the required information.

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

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

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

7. Click Install to begin the installation.

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

8. Click Finish.

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

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

More information:

Agent Error Messages on Windows (see page 163)
Display Debugging Information During Agent Installation (see page 146)

How to Install the Agent Using a Silent Installer

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

To install the agent using a silent installer

1. Configure the installer.properties file (see page 33).

2. Run the silent installer:

   - on UNIX (see page 42)
   - on Windows (see page 42)

3. Review the generated log file (see page 42).

Configure the installer.properties File

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

To configure the installer.properties file

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

   The properties are set in the installer.properties file.

Silent Installer Properties

The installer.properties file contains the following properties for the agent:

USER_INSTALL_DIR

Specifies the path to the location where you want to install the agent program files. The specified location must be empty.

Note: For Windows, use two backward slashes to separate directories in the path.

USER_SHORTCUTS

Specifies the path to the location where you want a shortcut to the agent to appear.

Default: C:\\Documents and Settings\\All Users\\Start Menu\\Programs\\CA

JVM_DOT

Specifies the full path to the JRE directory.

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

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

JVM_PATH

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

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

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

AGENT_INFO_1

Defines the agent name. You need the agent name when you configure the scheduling manager to work with the agent.

Default: AGENT

Limits: Up to 16 characters; the first character must be a letter

Notes:

- Agent names must begin with an alphabetic character and can contain any alphanumeric characters and the special characters @, $, and underscore (_). Because the scheduling manager uses agent names as file names, use standard file-naming conventions for your operating system.
- For CA Workload Automation DE, the agent name must be in uppercase.

AGENT_INFO_2

Specifies the main port number the agent uses to listen for incoming messages from the scheduling manager. You need this port when you configure the scheduling manager to work with the agent.

Default: 7520

Limits: 1024-65534

NUM_MANAGER_N=N

Specifies the number of scheduling managers (N) the agent works with.

Default: NUM_MANAGER_1=1

MANAGER_*n*_INFO_1

Specifies the name of the scheduling manager instance that the agent works with, where *n* is an integer that corresponds to the scheduling manager being configured.

Default: CENTRAL_MANAGER

Example: MYSERVER

MANAGER_*n*_INFO_2

Specifies the address of the scheduling manager that the agent works with, where *n* is an integer that corresponds to the scheduling manager being configured. This value corresponds to the IP address in the connection details for the scheduling manager. You can specify a list of addresses for the scheduling manager.

Example: 172.24.36.107 (IPv4) or 0:0:0:0:0:FFFF:192.168.00.00 (IPv6)

Notes:

- You can specify a DNS name instead of the IP address for the scheduling manager. However, your agent computer must be able to resolve the DNS name at all times. If there is a DNS outage and your agent computer cannot resolve DNS names, the agent cannot communicate with the scheduling manager.
- If the scheduling manager address never changes, enter the DNS name for the scheduling manager in the hosts file for your agent computer. This entry helps ensure that the IP address can be resolved after DNS disruptions.

MANAGER_*n*_INFO_3

Specifies the port that the scheduling manager listens on for communication from agents, where *n* is an integer that corresponds to the scheduling manager being configured. This value corresponds to the port number in the connection details for the scheduling manager.

Default: 7507

Limits: 1024-65534

STRONG_ENCRYPTION_CIPHER

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

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

Default: DES

STRONG_ENCRYPTION_KEYGEN

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

- AES—32 hexadecimal character encryption key.

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

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

DESEDE—48 hexadecimal character encryption key

STRONG_ENCRYPTION_FILE

Specifies the path to the text file that stores the encryption key for the agent.

LOCAL_SECURITY

Specifies whether local security on the agent is enabled or disabled. Local security on the agent controls which scheduling manager user IDs can perform certain actions, for example, which user IDs can issue CONTROL messages to the agent. If you enable local security, define security rules in a security.txt file.

- off—Disables local security.
- on—Enables local security.

Default: off

The following properties apply if you want to connect the agent to an SNMP manager.

SNMP_MGMT_CONN

Enables an SNMP connector that lets you use an SNMP Manager to monitor and control the agent. The agent supports SNMP v1, v2, and v3. This option requires the SNMP Manager address and UDP port.

- 0—Disables the SNMP connector
- 1—Enables the SNMP connector

MGMT_SNMP_HOST

Identifies the SNMP Manager IP address or DNS name. Your SNMP administrator can provide the host name.

Default: localhost

Example: 172.24.36.107

MGMT_CONN_AGENT_PORT

Specifies the SNMP GET/SET listening port.

Default: 161

Limits: 1-65535

JMX_PLUGIN

Enables a JMX connector that lets you use a JMX console to monitor and control the agent. You can use any JMX console that implements JSR-160.

- 0—Disables the JMX connector
- 1—Enables the JMX connector

JMX_CONNECTOR_PORT

Specifies the port where the JMX connector listens.

Default: 1099

Limits: 1-65534

The following FTP properties apply if you want to configure the agent to run FTP workload (see page 121).

FTP_PLUGIN

Enables the FTP plug-in on the agent, which lets you configure FTP client and FTP server options.

- 0—Disables the FTP plug-in
- 1—Enables the FTP plug-in

Default: 0 (disabled)

FTP_SSL_CLIENT_ENABLED

Specifies whether the agent can act as an FTP client using Regular or Secure Sockets Layer (SSL) encryption for FTP transfers.

- false—Enables regular encryption
- true—Enables SSL encryption

Default: false

FTP_NO_SERVER

Sets whether the agent can act as an FTP server.

- false—Enables FTP server
- true—Disables FTP server

Default: false

FTP_SSL_SVR_ENABLED

Specifies whether the agent can act as an FTP server using regular or Secure Sockets Layer (SSL) encryption for FTP transfers.

- false—Enables regular encryption
- true—Enables SSL encryption

Default: false

FTP_SVR_PORT

Specifies the port number for the agent to act as an FTP server.

Default: 21

Limits: 1-65534

FTP_USER_ID

Specifies the FTP user ID required to access the FTP server.

FTP_PASSWORD

Specifies the password corresponding to the FTP user ID.

Limits: case-sensitive

FTP_PASSWORD_V

Confirms the FTP password.

The following SNMP properties apply if you want to configure the agent as an SNMP Manager (see page 93).

SNMP_PLUGIN

Enables the agent to act as an SNMP Manager to emit and listen for SNMP traps. This option lets users define and run SNMP job types. The agent supports SNMP v1, v2, and v3.

- 0—Disables the SNMP plug-in
- 1—Enables the SNMP plug-in

Default: 0 (disabled)

SNMP_P_TRAP_PORT

Specifies the agent port listening for trap information.

Default: 162

Limits: 1-65535

CA_SOCKET

Enables a connection to CA Workload Automation AE using the CA Secure Socket Adapter (SSA).

- 0—Disables the CA Secure Socket Adapter
- 1—Enables the CA Secure Socket Adapter

Default: 0 (disabled)

SSA_COMPONENT_PATH

Specifies the path to the *.so file (UNIX) or *.dll file (Windows) for communication with CA Workload Automation AE using the CA Secure Socket Adapter.

Default:

- For UNIX:

  ```
  /opt/CA/SharedComponents/Csam/SockAdapter/lib
  ```

- For Windows:

  ```
  C:\Program Files\CA\SharedComponents\Csam\SockAdapter\bin
  ```

Note: For UNIX systems, also specify the EWA_SHARED_DIR property.

EWA_SHARED_DIR

Specifies the path to the shared components directory for CA Workload Automation AE.

Default: /opt/CA/SharedComponents

Note: This property is required for UNIX system.

NUM_MANAGER_VARS_2

Specifies the number of manager environment variables for the scheduling manager.

Limits: 0-3

MANAGER_VARS_*n*_INFO_1

Specifies the name of the specific scheduling manager the environment variables apply to, where *n* is an integer that corresponds to the scheduling manager being configured.

Example: MANAGER1_VAR

MANAGER_VARS_n_INFO_2

Specifies the path to the file that stores the environment variables, where *n* is an integer that corresponds to the scheduling manager being configured.

Example: C:\\MANAGER_1\\FILE1.TXT

NUM_USER_VARS_2

Specifies the number of user environment variables for the scheduling manager.

Limits: 0-3

USER_VARS_*n*_INFO_1

Specifies the name of the user the environment variables apply to, where *n* is an integer that corresponds to the scheduling manager being configured.

Example: USER1

USER_VARS_*n*_INFO_2

Specifies the path to the file that defines user-specific variables.

Example: C:\\USER1\\FILE1.TXT

LOOKUPCMD

Determines how to specify the script or command name (UNIX) or command file (Windows) to run in a job definition.

- false—The full path to the script, command name, or command file must be specified in the job definition.
- true—The script, command name, or command file can be specified without the full path in a job definition. On UNIX, the agent looks up the path to the script or command name for the specified user ID. On Windows, the agent looks up the path to the command file in the following directories, in the order listed: agent installation directory, WINDOWS\system32 directory on 32-bit versions of Windows (or WINDOWS\SysWow64 directory on 64-bit versions of Windows), WINDOWS\system directory, WINDOWS directory, the system path and user path.

Default: true

Notes:

- If set to true, verify that the agent on UNIX is running under the root account.
- The agent does not resolve environment variables specified in the command file path for Windows jobs.

JOBLOG

Sets whether the agent creates a job log for each job that runs.

- false—Disables job logs
- true—Enables job logs

Default: true

WIN_SERVICE_*n*

Specifies the name for an agent, installed on Windows, that appears in the list of Services, where *n* is an integer that corresponds to the scheduling manager being configured. You can control the agent as a Windows Service.

Default: CA Workload Automation Agent 11.3

Silent Installer Example

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

Example: Configuring the installer.properties File

The properties file in this example installs an agent named AGENT2 on a UNIX system in the agent_solaris_aes directory. The agent uses port 34520 for communication with the scheduling manager named manager1 that has an IP address of ::FFFF:192.168.00.00 and uses port 8507. Local security for AGENT2 is enabled. The agent uses the AES cipher algorithm. Environment variables used by the agent and scheduling manager are located in the FILE1.txt file.

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

Run the Silent Installer on UNIX

Run the silent installer to perform the agent installation.

To run the silent installer on UNIX

Type the following command at the command prompt:

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

path

Specifies the path to the installer.properties file.

The agent is installed.

Run the Silent Installer on Windows

Run the silent installer to perform the agent installation.

To run the silent installer on Windows

Type the following command at the command prompt:

```
setup.exe -f installer.properties
```

The agent is installed.

Review the Generated Log File

The agent installation program creates a log file, which you can review for installation errors. Review the following file located in the agent installation directory:

```
CA_Workload_Automation_Agent_R11.3_InstallLog.log
```

Migrating an R6 or R7 ESP System Agent to R11.3

You can use the r11.3 agent installation program to migrate a Release 6 or Release 7 ESP System Agent to r11.3. Migrating an existing agent to the new release lets you preserve your agent settings. The installation program installs new binaries and converts the existing agentparm.txt file and other artifacts to r11.3. You can migrate an existing R6 or R7 agent using the interative program (see page 43) or command-based silent installer (see page 44).

Note: The r11.3 agent installation program does not support a full upgrade. After migrating an agent, you can configure agent parameters to enable the r11.3 features such as the management connectors and enhanced encryption.

More information:

Configure the Agent to Connect with a JMX Console (see page 101)
Configure the Agent to Connect with an SNMP Manager (see page 102)
Configure the Agent for Encryption Standard FIPS 140-2 (see page 111)

Convert an Existing agentparm.txt File Using an Interactive Program

You can convert your Release 6 or Release 7 agentparm.txt file to r11.3 to preserve your existing settings. You can use the interactive installation program to do the conversion.

To convert an existing agentparm.txt file using the interactive program

1. Run the interactive installation program.
   - On UNIX (see page 30)
   - On Windows (see page 31)

   The agent installation program opens.
2. Accept the license agreement and enter the required information until you are prompted for the AgentParm File Conversion.
3. Select Yes.
4. Continue to the next prompt.
5. Enter the path to your existing agentparm.txt file.
6. Complete the installation.

   The agentparm.txt file is converted to r11.3.

Convert an Existing agentparm.txt File Using the Silent Installer

You can convert your Release 6 or Release 7 agentparm.txt file to r11.3 to preserve your existing settings. You can use the silent installer to do the conversion.

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

1. Copy the r11.3 installer.properties file to your agent computer. The file is available on the product CD or CA Support Online website, found at http://ca.com/support.

2. Open the installer.properties file you copied.

3. Disable the following property by adding a comment (#) character:

   ```
   #AGENTPARM_CONVERT_2 =No
   ```

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

   ```
   AGENTPARM_CONVERT_1 =Yes
   ```

   Enabling this property preserves the agentparm.txt settings of your existing agent.

5. Enable and edit the following property to specify the path to the agentparm.txt file for your existing agent:

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

   For example, if you have a Release 6 ESP System Agent installed in C:\\Program Files\\Cybermation\\ESP System Agent R6\\agentparm.txt, edit the property as follows:

   ```
   OLD_AGENT_PARM=C:\\Program Files\\Cybermation\\ESP System Agent
   R6\\agentparm.txt
   ```

6. Verify that all other properties are disabled by adding the comment (#) character to each property that is uncommented.

7. Save the file.

   The properties are set in the installer.properties file.

8. Run the silent installer:

   - on UNIX (see page 42)
   - on Windows (see page 42)

9. Review the generated log file (see page 42).

How to Remove the Agent

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

To remove the agent, follow these steps:

1. Uninstall the agent:
   - on UNIX (see page 45)
   - on Windows (see page 46)
2. Remove the agent from the scheduling manager.

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

Uninstall the Agent on UNIX

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

To uninstall the agent on UNIX

1. Verify that all workload is complete.
2. Stop the agent.
3. Delete the agent installation folder.

   The agent is uninstalled.

Uninstall the Agent on Windows

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

To uninstall the agent on Windows

1. Verify that all workload is complete.
2. Stop the agent.
3. Click Start, Program menu to uninstall or run the uninstall file located in *agentinstall directory*\UninstallData.

   The Installation Type dialog opens.
4. Click Uninstall.

   The Uninstall Complete dialog opens when the uninstall is finished.
5. Click Done.

   The agent wizard closes.

Chapter 4: Controlling the Agent

This section contains the following topics:

Starting the Agent

Depending on your operating system, you have several options for starting the agent.

On UNIX, you issue a command to run a start script.

On Windows, you can start the agent as a Windows service or use the command prompt. You can also set the agent on Windows to start automatically each time you restart the system.

Start the Agent on UNIX

You issue a command to run a script that starts the agent.

Important! Use the root account to install and start the agent on UNIX. Starting the agent as root enables you to run jobs under different user accounts. If you start the agent with an account other than root, the agent cannot switch users. You cannot run jobs under different user accounts when the agent is not running under root. If you plan to use an account other than root to start the agent, verify that the agent is installed using the same user account to avoid permission problems. Also verify that the account has the permissions to run the commands and scripts that run on the agent computer.

To start the agent on UNIX

1. Verify that the cybAgent process and related Java processes from the previous run of the agent were shut down correctly.
2. Change to the agent installation directory.
3. Enter the following command:

   ```
   ./cybAgent &
   ```

   The agent runs in the background.

More information:

Agent Will Not Start—cybAgent Script is Missing (see page 166)

Start the Agent on Windows Using the Command Prompt

By default, the startup type for the agent service is manual. You can start the agent service manually using the command prompt.

To start the agent on Windows using the command prompt

1. Change to the agent installation directory.
2. Enter *one* of the following commands:
   - cybAgent -a
   - net start *Service_Name*

   Service_Name

   Specifies the value defined by the oscomponent.servicedisplayname parameter in the agentparm.txt file.

   The agent starts.

Start the Agent as a Windows Service

By default, the startup type for the agent service is manual. You can start the agent service manually using the Windows Control Panel.

To start the agent as a Windows Service

1. Click Start, Control Panel.

   The Control Panel dialog opens.
2. Double-click Administrative Tools.

   The Administrative Tools dialog opens.
3. Double-click Services.

   The Services dialog opens.
4. Right-click the agent service and click Start.

   The agent starts.

Set the Agent on Windows to Start Automatically

By default, the startup type for the agent service is manual; the agent does not start on system startup. You can start the agent manually using the Control Panel or command prompt. You can set the Service Startup Type as Automatic to start the service at system startup.

To set the agent on Windows to start automatically

1. Click Start, Control Panel.

   The Control Panel dialog opens.

2. Double-click Administrative Tools.

   The Administrative Tools dialog opens.

3. Double-click Services.

   The Services dialog opens.

4. Right-click the agent service and select Properties.

   A properties dialog opens for the agent service.

5. Select Automatic from the Startup type drop-down list on the dialog.

6. Click OK.

   The agent is set to start automatically the next time it starts up.

Run the Agent in Windows Console Mode

You can run the agent in console mode using the cybAgent command with the -debug extension. The agent runs as a normal console application. Running the agent in console mode is useful when diagnosing a problem. In this mode, the agent runs in the foreground. Because some applications cannot run correctly while the agent runs as a service, try to run these applications using console mode. To keep the agent running, do not close the DOS window you used to start it.

To run the agent in Windows console mode

1. Stop the agent.

2. Change to the agent installation directory.

3. Enter the following the command:

   ```
   cybAgent -debug
   ```

   The agent runs in the foreground.

Note: You can run multiple agents in console mode.

Stopping the Agent

Depending on your operating system, you have several options for stopping the agent.

On UNIX, you issue a command to run a stop script.

On Windows, you can stop the agent as a Windows service or using the command prompt.

Stop the Agent on UNIX

You issue a command to run a script that stops the agent.

To stop the agent on UNIX

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

   ```
   ./cybAgent -s
   ```

   The agent stops.

Stop the Agent on Windows Using the Command Prompt

You can stop the agent service manually using the command prompt.

To stop the agent on Windows using the command prompt

1. Change to the agent installation directory.
2. Enter one of the following commands:
   - cybAgent -s
   - net stop *Service_Name*

   Service_Name

   Specifies the value defined by the oscomponent.servicedisplayname parameter in the agentparm.txt file.

   The agent stops.

Stop the Agent as a Windows Service

You can stop the agent service manually using the Windows Control Panel.

To stop the agent on Windows using the Control Panel

1. Click Start, Control Panel.

   The Control Panel dialog opens.

2. Double-click Administrative Tools.

   The Administrative Tools dialog opens.

3. Double-click Services.

   The Services dialog opens.

4. Right-click the agent service and click Stop.

   The agent stops.

Stop the Agent Running in Windows Console Mode

You can stop an agent that is using console mode to run in the foreground.

To stop the agent on Windows in console mode, press Ctrl+C.

Verifying the Status of the Agent

If there are problems running workload using the agent, you can verify whether the agent is running or has stopped. You can verify the status in the following ways:

- Verify the status file—The status file shows whether the agent is running or a controlled shutdown has occurred.
- Verify the agent process—The agent process status can verify whether the agent is down, which is helpful when a controlled shutdown did not occur and the status file does not show the correct status.

View the status File

The status file describes the status of the agent core. To verify that the agent is running or a controlled shutdown has occurred, view the status file.

To view the status file

1. Change to the agent installation directory.
2. Do *one* of following actions depending on your system:
   - On UNIX, enter the cat status command.
   - On Windows, open the status file using a text editor.

Check the Agent Process Status on UNIX

If the agent is down and a controlled shutdown did not occur, check the agent process status.

To check the agent process status on UNIX

1. Change to the agent installation directory.
2. Enter one of the following commands:
   - To check a single agent, enter the **# ps -ef | grep PID** command.
   - To check multiple agents or an agent with an unknown PID, enter **# ps -ef | grep cybAgent**.
   - To check a separate Java process before a new start, enter **# ps -ef | grep Java**.

Example: Check the Process Status of a Single Agent on UNIX

Suppose that you want to check the process status of an agent that has the PID number 13214.

To check the agent process status on UNIX

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

```
# ps -ef | grep 13214
```

Check the Agent Status on Windows

If the agent is down and a controlled shutdown did not occur, check the agent status on Windows using the Control Panel.

To check the agent status on Windows

1. Click Start, Control Panel.

   The Windows Control Panel opens.

2. Double-click Administrative Tools.

   The Administrative Tools dialog opens.

3. Double-click Services.

   The Services dialog opens.

4. Locate the agent service name and check the Status column.

Chapter 5: Configuring the Agent

Note: The UNIX instructions in this document also apply to Linux systems unless otherwise noted.

This section contains the following topics:

How to Configure Agent Parameters

You configure agent parameters by editing the agentparm.txt file, 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:

1. Configure agent parameters on the agent (see page 56).
2. Configure agent parameters on the scheduling manager (see page 56).

Configure Agent Parameters on the Agent

Use the following procedure to configure agent parameters on CA WA Agent for UNIX, Linux, or Windows.

To configure agent parameters on the agent

1. Change to the agent installation directory.
2. Stop the agent. At the command prompt, enter the following command:
   - On UNIX:
     ```
     ./cybAgent -s
     ```
   - On Windows:
     ```
     cybAgent -s
     ```

   The agent stops.
3. Open the agentparm.txt file located in the agent installation directory.
4. Edit the parameters to make the required changes.
5. Save and close the agentparm.txt file.
6. Start the agent. At the command prompt, enter the following command:
   - On UNIX:
     ```
     ./cybAgent &
     ```
   - On Windows:
     ```
     cybAgent -a
     ```

   The agent starts and the parameters are configured.

More information:

Agent Parameters in the agentparm.txt File (see page 57)
Agent Parameters used for Troubleshooting (see page 170)

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.

Agent Parameters in the agentparm.txt File

The following agent parameters appear in the agentparm.txt file in the order listed. The parameter values are set during the agent installation. You can modify these parameters as required. To configure the agent for additional functions, follow the procedures in this chapter.

log.level

Specifies the type and number of logs the agent generates. This parameter is important for troubleshooting.

- 0, 1, or 2—Creates logs for any errors including the receiver and transmitter logs. Level 2 is adequate for production, unless problems arise requiring more details for troubleshooting.
- 3—Adds queues. If this value is specified, the agent ignores the log.maxsize parameter.
- 4 or 5—Adds debugging information. Use log level 5 for setup and initial testing.
- 6-8—Adds tracing information to diagnose a problem. These levels are not intended for continuous use.

Default: 5

Example: log.level=2

log.archive

Defines the log archiving options:

- 0—Appends current date and time to the log file.
- 1—Renames to logfile.archive and starts a new file.
- 2—Removes current file.
- 3—Appends new log entries to the current logs.

Default: 0

log.maxsize=*maximum size*[B|K|M|G]

Specifies the maximum log size. When the log file exceeds the specified size, the agent archives it and starts a new log file. If the log.archive parameter is set to three, the agent ignores this parameter. The agent does not create an archive file, but it does append all logs. You can specify the following optional modifiers:

- B—Specifies the size in bytes.
- K—Specifies the size in kilobytes.
- M—Specifies the size in megabytes.
- G—Specifies the size in gigabytes.

Note: The default (no modifier) size is in bytes.

Limits: 2G

Default: 1M

agentname

Defines the agent name. You need the agent name when you configure the scheduling manager to work with the agent.

Default: AGENT

Limits: Up to 16 characters; the first character must be a letter

Notes:

- Agent names must begin with an alphabetic character and can contain any alphanumeric characters and the special characters @, $, and underscore (_). Because the scheduling manager uses agent names as file names, use standard file-naming conventions for your operating system.
- For CA Workload Automation DE, the agent name must be in uppercase.

communication.inputport

Specifies the main port number the agent uses to listen for incoming messages from the scheduling manager. You need this port when you configure the scheduling manager to work with the agent.

Default: 7520

Limits: 1024-65534

Note: On UNIX, ports 1–1023 are reserved ports, which require root access.

communication.receiver.socket.main

Specifies the type of socket the agent uses for its main port. The value of this parameter must be different from the communication.receiver.socket.aux parameter. You can specify the following socket types:

- plain
- dylan

Default: plain

Note: CA Workload Automation DE does not require this parameter. z/Linux systems use plain socket types only.

communication.managerid_*n*

Specifies the name of the scheduling manager instance that the agent works with, where *n* is an integer that corresponds to the scheduling manager being configured.

Default: CENTRAL_MANAGER

Example: MYSERVER

communication.manageraddress_*n=address 1;...;address_m*

Specifies the address of the scheduling manager that the agent works with, where *n* is an integer that corresponds to the scheduling manager being configured. This value corresponds to the IP address in the connection details for the scheduling manager. You can specify a list of addresses for the scheduling manager.

Example: 172.24.36.107 (IPv4) or 0:0:0:0:0:FFFF:192.168.00.00 (IPv6)

Notes:

- You can specify a DNS name instead of the IP address for the scheduling manager. However, your agent computer must be able to resolve the DNS name at all times. If there is a DNS outage and your agent computer cannot resolve DNS names, the agent cannot communicate with the scheduling manager.
- If the scheduling manager address never changes, enter the DNS name for the scheduling manager in the hosts file for your agent computer. This entry helps ensure that the IP address can be resolved after DNS disruptions.

communication.managerport_*n*

Specifies the port that the scheduling manager listens on for communication from agents, where *n* is an integer that corresponds to the scheduling manager being configured. This value corresponds to the port number in the connection details for the scheduling manager.

Default: 7507

Limits: 1024-65534

communication.monitorobject_*n*

Specifies the monitor object for the scheduling manager that is used in agent alive ping.

communication.socket_*n*

Defines the socket type the agent and scheduling manager use for communication, where *n* is an integer starting at one that corresponds to the scheduling manager being configured. The following socket types are available:

- plain
- dylan

Default: plain

Note: z/Linux systems use plain socket types only.

security.filename

Specifies the path to the security file that contains the security rules that define local security on the agent.

Default: *agentinstallDir*/security.txt

security.level

Specifies whether local security on the agent is enabled or disabled. Local security on the agent controls which scheduling manager user IDs can perform certain actions, for example, which user IDs can issue CONTROL messages to the agent. If you enable local security, define security rules in a security.txt file.

security.cryptkey

Specifies the path to the text file that stores the encryption key for the agent.

Default:

- For UNIX:

  ```
  /CA/WA_Agent_R11_3/cyrptkey.txt
  ```

- For Windows:

  ```
  C:\Program Files\CA\WA Agent R11.3\cryptkey.txt
  ```

initiators.class_n=*jobclass*,*number of initiators*

Describes job classes and the number of initiators that can process jobs that are assigned a particular job class. Use a new line for each initiators.class_*n* parameter, where *n* is an integer starting at the value 1. By controlling the type and number of initiators, you can have greater control over the initiation of jobs and manually balance the loads on system resources.

The parameter initiators.afmjobclassmap_*n* relates to this parameter. However, the value of *n* does not have to match in both parameters.

For UNIX workload, depending on the number of initiators you assign, you may need to increase the number of threads that can be run per process on your operating system.

Examples:

initiators.class_1=Default,1000

initiators.class_2=POJO,100

core.health.monitor.enable

Specifies whether resource usage information within the Java Virtual Machine (JVM), such as memory usage and threads information, must be logged.

- false—Does not log resource usage information within the JVM
- true—Logs the resource usage information within the JVM to a file named simple_health_monitor.log in the log folder

Default: true

Note: The log.level parameter must be set to 5 or greater to log this information.

spooldir

Specifies the path to the spool file directory.

Default: spool subdirectory of the agent installation directory

oscomponent.javapath

Specifies the full path to the directory where Java resides.

oscomponent.jvm

Specifies the Java virtual machine (JVM) to use.

plugins.start_internal_*n*

Specifies the agent plug-in to start by the core Java agent.

n

Specifies an integer, assigned to the agent plug-in, starting at 1. The *n* suffix must increase sequentially for each agent plug-in.

oscomponent.classpath

Specifies the path to jar files required by the agent.

management.snmp.mibfile

Specifies the path to the MIB file that describes the metrics and SNMP traps for the agent.

Default: *agentinstalldir*/cybermation.mib

management.snmp.host

Identifies the SNMP Manager IP address or DNS name. Your SNMP administrator can provide the host name.

management.snmp.port

Specifies the SNMP Manager UDP port. Your SNMP administrator can provide this port number.

Default: 162

Limits: 1-65535

management.snmp.community

Specifies the type of network the SNMP traps are sent across for SNMP v1 or v2 only. Your SNMP administrator can provide the type.

- public—Identifies an unsecured network, for example, the Internet.
- private—Identifies a secure network, for example, a local area network.

Default: public

ftp.noserver

Specifies whether the agent FTP server is enabled or disabled. If ftp.noserver is set to false, the FTP server is enabled. If the ftp.noserver is set to true, the FTP server is disabled.

Default: true

ftp.serverport

Specifies the port number for the agent to act as an FTP server.

Default: 21

Limits: 1-65534

ftp.client.ssl

Specifies whether all FTP jobs on the agent computer automatically use SSL communication.

- false—Disables SSL communication.
- true—Enables SSL communication.

ftp.client.ssl.truststore

Specifies the full path name of the truststore file. The default file name is cacerts. You can use keytool, provided with the JRE, to create your own truststore.

ftp.client.ssl.truststore.password

Specifies the encrypted password for the client truststore file, for example, cacerts, that contains some common CA X509 certificates.

Default: changeit (encrypted)

Note: You can use the agent password utility to encrypt your password before using it in the agentparm.txt file.

ftp.server.ssl

Specifies that the FTP server handles both non-SSL and SSL FTP.

ftp.server.ssl.keystore.password

Specifies the encrypted password for the server keystore that contains an X509 certificate. This password is sent to the client during the handshake process.

Default: cyberuser (encrypted)

management.connector_*n*

Identifies the type of management connector the agent uses to connect to an external application, where *n* is an integer starting from 1. You can specify the following types of connectors:

- jmx—Specifies a JMX connector, built into the agent, that lets you use a JMX console to monitor and control the agent.
- snmp—Specifies an SNMP connector, built into the agent, that lets you use an SNMP manager to monitor and control the agent.

management.jmx.port

Specifies the port where the JMX connector listens.

Default: 1099

Limits: 1-65534

oscomponent.servicename

Specifies the agent name as it appears in the list of services. The length of the service display name must fall within Windows guidelines. Use a unique name if you install more than one agent.

Default: CA Workload Automation Agent 11.3

oscomponent.servicedisplayname

Specifies the name for an agent installed on Windows that appears in the list of Services. You can control the agent as a Windows Service.

Default: CA Workload Automation Agent 11.3

oscomponent.loginshell

Indicates how to invoke the Shell program when executing a script.

- false—The agent ignores the shell as a login shell.
- true—The agent invokes the shell as a login shell when you specify true. The shell program looks in the directory specified by the HOME environment variable and tries to execute the login scripts of the user (in addition to the .cshrc script).

Default: false

Note: For most systems, this parameter affects only the C and Korn shells. The Bourne shell ignores the oscomponent.loginshell parameter.

oscomponent.defaultshell

Identifies the shell in which scripts are run on the system.

Default: /bin/sh

oscomponent.validshell

Identifies the full path and name of every shell that is valid for use on the agent. Separate each shell with a comma.

Default: /usr/bin/sh,/bin/csh,/bin/ksh,/bin/sh,/bin/bash

Note: This parameter is selected when the oscomponent.checkvalidshell parameter is set to true (the default). If the shell used in a job definition or script is not specified in this parameter, the job fails.

oscomponent.checkvalidshell

Determines whether the agent checks valid shells.

- false—The agent bypasses the valid shell check.
- true—The agent checks valid shells. All shells that jobs use must be specified in the oscomponent.validshell parameter.

Default: true

oscomponent.lookupcommand

Determines how to specify the script or command name (UNIX) or command file (Windows) to run in a job definition.

- false—The full path to the script, command name, or command file must be specified in the job definition.
- true—The script, command name, or command file can be specified without the full path in a job definition. On UNIX, the agent looks up the path to the script or command name for the specified user ID. On Windows, the agent looks up the path to the command file in the following directories, in the order listed: agent installation directory, WINDOWS\system32 directory on 32-bit versions of Windows (or WINDOWS\SysWow64 directory on 64-bit versions of Windows), WINDOWS\system directory, WINDOWS directory, the system path and user path.

Default: true

Notes:

- If set to true, verify that the agent on UNIX is running under the root account.
- The agent does not resolve environment variables specified in the command file path for Windows jobs.

oscomponent.joblog

Sets whether the agent creates a job log for each job that runs.

- false—Disables job logs
- true—Enables job logs

Default: true

Configure Communication with a Scheduling Manager

You can change your scheduling manager connection information or add a connection to a different scheduling manager. If you are using the agent with two scheduling managers that require different socket types for communication, you can specify a main and auxiliary socket for the agent.

To configure communication with a scheduling manager, configure the following agent parameters on the agent:

communication.inputport

Specifies the main port number the agent uses to listen for incoming messages from the scheduling manager. You need this port when you configure the scheduling manager to work with the agent.

Default: 7520

Limits: 1024-65534

communication.inputport.aux

Optional. Specifies the auxiliary port number the agent uses to listen for incoming messages from the scheduling manager.

communication.manageraddress_*n*

Specifies the address of the scheduling manager that the agent works with, where *n* is an integer that corresponds to the scheduling manager being configured. This value corresponds to the IP address in the connection details for the scheduling manager. You can specify a list of addresses for the scheduling manager.

Example: 172.24.36.107 (IPv4) or 0:0:0:0:0:FFFF:192.168.00.00 (IPv6)

Notes:

- You can specify a DNS name instead of the IP address for the scheduling manager. However, your agent computer must be able to resolve the DNS name at all times. If there is a DNS outage and your agent computer cannot resolve DNS names, the agent cannot communicate with the scheduling manager.
- If the scheduling manager address never changes, enter the DNS name for the scheduling manager in the hosts file for your agent computer. This entry helps ensure that the IP address can be resolved after DNS disruptions.

communication.managerid_*n*

Specifies the name of the scheduling manager instance that the agent works with, where *n* is an integer that corresponds to the scheduling manager being configured.

Default: CENTRAL_MANAGER

Example: MYSERVER

communication.managerport_*n*

Specifies the port that the scheduling manager listens on for communication from agents, where *n* is an integer that corresponds to the scheduling manager being configured. This value corresponds to the port number in the connection details for the scheduling manager.

Default: 7507

Limits: 1024-65534

communication.monitorobject_*n*

Specifies the monitor object for the scheduling manager that is used in agent alive ping.

communication.receiver.socket.aux

Specifies the type of socket the agent uses for its auxiliary port. The value of this parameter must be different from the communication.receiver.socket.main parameter. You can specify the following socket types:

- plain
- dylan

Note: CA Workload Automation DE does not require this parameter.

communication.receiver.socket.main

Specifies the type of socket the agent uses for its main port. The value of this parameter must be different from the communication.receiver.socket.aux parameter. You can specify the following socket types:

- plain
- dylan

Default: plain

Note: CA Workload Automation DE does not require this parameter.

communication.socket_*n*

Defines the socket type the agent and scheduling manager use for communication, where *n* is an integer starting at one that corresponds to the scheduling manager being configured. The following socket types are available:

- plain
- dylan

Default: plain

Note: CA Workload Automation DE does not require this parameter.

Note: You can configure the agent to work with multiple scheduling managers by adding additional definitions in the agentparm.txt file.

Example: Configure the Agent to Communicate with a Scheduling Manager

In this example, the following configuration parameters are set in the agentparm.txt file for a scheduling manager running under the instance “ACE” at address 130.200.146.134. The scheduling manager listens for incoming messages from the agent on port 49155:

```
communication.inputport=7520
communication.managerid_1=ACE
communication.manageraddress_1=130.200.146.134
communication.managerport_1=49155
communication.monitorobject_1=CAEWA_AGENT/AGENTMON1.0/MAIN
communication.receiver.socket.main=plain
communication.socket_1=plain
```

More information:

Agent Parameters in the agentparm.txt File (see page 57)

Configure the Agent for Internet Protocol Version 6 (IPV6) Communication

If your scheduling manager uses Internet Protocol Version 6 (IPV6), configure the agent for this protocol also.

To configure the agent for IPV6 communication, configure the following agent parameters on the agent:

```
java.net.preferIPv6Addresses=true
java.net.preferIPv4Stack=false
```

More information:

Configure Agent Parameters on the Agent (see page 56)

How to Configure SSA Communication with CA Workload Automation AE

Notes:

- This procedure only applies to agent installations that communicate with CA Workload Automation AE and that support SSA.
- By default the agent uses plain socket ports for communication. Although you can change to SSA communication, we do not recommend it.

CA Secure Socket Adapter (SSA) lets CA Workload Automation AE and the agent use a single multiplexed communication port to ease firewall administration.

To configure SSA communication with CA Workload Automation AE, follow these steps:

1. Install CA Secure Socket Adapter (SSA):
   - Install SSA on UNIX (see page 69).
   - Install SSA on Windows (see page 70).
2. Configure the agent to communicate using SSA ports (see page 71).
3. Define the agent SSA port on CA Workload Automation AE (see page 73).
4. Test communication between CA Workload Automation AE and the Agent.

   For more information about testing communication, see the *CA Workload Automation AE UNIX Implementation Guide* or *CA Workload Automation AE Windows Implementation Guide*.

Install CA Secure Socket Adapter (SSA) on UNIX

CA Secure Socket Adapter (SSA) is a CA common component that is provided with CA Workload Automation AE. SSA lets you enable port multiplexing, which restricts communication to a single physical port. For more information about SSA, see the *CA Workload Automation AE UNIX Implementation Guide*.

You can launch the installation program using X-terminal or using console mode.

To install CA Secure Socket Adapter (SSA) on UNIX

1. Set the $DISPLAY environment variable to use X-terminal or unset it to use console mode.
2. Mount the CA Common Components media.
3. Run the following command:

   ```
   ./ccc_setup.sh
   ```

   Note: The setup script performs various system checks, which can take a few minutes.

   After the system checks are complete, the CA Common Components panel appears with Next selected.
4. Press Enter.

   The License Agreement panel appears.
5. Accept the license agreement.

   Note: In console mode, press the Tab key to select the scroll bar, and press the plus key (+) to scroll through the page and read the license text. If you agree with the license agreement, select I Agree.

   The Component Selection panel appears.
6. Select CA Secure Socket Adapter.

   Note: In console mode, press the down-arrow key (or Tab key) to move the cursor through the list of components, and use the spacebar to select or clear an item.
7. Select Next.

   The Installation Path panel appears.
8. Enter the path to the directory where you want to install SSA, and select Next.

   The Review Settings panel appears.

9. Review the information and, if it is correct, select Next.

   The installation process begins and the progress is displayed. When the installation completes, you are prompted to exit the installation.

10. Select Finish.

11. Log out and then log back in to reset the environment.

    The CA Secure Socket Adapter is installed.

Install CA Secure Socket Adapter (SSA) on Windows

CA Secure Socket Adapter (SSA) is a CA common component that is provided with CA Workload Automation AE. SSA lets you enable port multiplexing, which restricts communication to a single physical port. For more information about SSA, see the *CA Workload Automation AE Windows Implementation Guide*.

To install CA Secure Socket Adapter (SSA) on Windows

1. Insert the CA Common Components media into the DVD drive of the computer where you want to install the agent.

   The CA Common Components 11.3 Product Explorer dialog opens.

2. Expand CA Common Components 11.3 for Windows, select Common Components, and click Install.

   The Introduction panel appears.

3. Click Next.

   The License Agreement appears.

4. Accept the license agreement, and click Next.

   The Choose Install Set panel appears.

5. Select Custom, and click Next.

   By default, the CA Secure Socket Adapter, CA Embedded Entitlements Manager, and Common Communications Interface components are selected to be installed.

6. Unselect the CA Embedded Entitlements Manager and Common Communications Interface components, and click Next.

   The Get CA Common Components Install Directory panel appears.

7. Enter the path to the directory where you want to install SSA, and select Next.

   The Review Settings panel appears.

8. Review the information and, if it is correct, click Install.

   The CA Secure Socket Adapter is installed.

Configure the Agent to Communicate Using an SSA-Enabled Port

By default, the agent is configured to use plain socket ports for communication. To enable SSA communication between the agent and CA Workload Automation AE, you configure the agent to use Dylan socket ports. You use the csamconfigedit utility, installed with SSA, to specify the SSA port number and enable port multiplexing.

Important! The input communication port defined for the agent (communication.inputport) must match the port in the machine definition for the agent configured on CA Workload Automation AE.

To configure the agent to communicate using an SSA-enabled port

1. Enter the following command:

   ```
   csampmux stop
   ```

   The csampmuxf process stops.

2. Enter the following commands:

   ```
   cd $CSAM_SOCKADAPTER/bin
   csamconfigedit Port=value EnableSSL=False EnablePmux=True
   ```

   value

   Specifies the SSA port number of the agent. This port must not be in use by another application.

3. Change to the agent installation directory.
4. Stop the agent.
5. Open the agentparm.txt file.
6. Edit the following parameter:

   ```
   communication.inputport=port
   ```

   port

   Specifies the SSA port number configured using the csamconfigedit command.

7. Set the following parameter to change the socket type, as follows:

   ```
   communication.receiver.socket.main=dylan
   ```

8. Edit the oscomponent.classpath parameter, as follows:

   ```
   oscomponent.classpath=jars/*.jar;jars/ext/*;common_components_installation_path/Csam/SockAdapter/bin/casocket.jar
   ```

   common_components_installation_path

   Specifies the path to the directory where the CA common components are installed.

   UNIX Default: /opt/CA/SharedComponents

   Windows Default: C:\Program Files\CA\SC

   Note: Append the location of the casocket.jar file to the classpath to specify the location of CA Secure Socket Adapter.

9. Save and close the agentparm.txt file.

10. Start the agent.

    The agent is configured to communicate using an SSA-enabled port.

Define the Agent SSA Port on CA Workload Automation AE

To communicate with the agent using an SSA-enabled port, you must change the port defined in the machine definition for the agent on CA Workload Automation AE.

To define the agent SSA port on CA Workload Automation AE

1. Specify one of the following subcommands in a JIL script or at the jil command line:

   - If you are creating a new agent machine definition:

     ```
     insert_machine: machine_name
     ```

   - If you are updating an existing agent machine definition:

     ```
     update_machine: machine_name
     ```

   machine_name

   Specifies the name of the agent defined on CA Workload Automation AE.

2. Specify the following attribute:

   ```
   port: port_number
   ```

   port_number

   Specifies the port that CA Workload Automation AE uses to listen for traffic. If you configured SSA on the agent, this value is the agent port number configured using the csamconfigedit command. This value must match the communication.input port parameter in the agentparm.txt file for the agent.

   The SSA port is defined in the machine definition for the agent on CA Workload Automation AE.

   Note: For more information about JIL subcommands and attributes, see the *CA Workload Automation AE Reference Guide*.

Define a Default User ID

To run a job under a user ID, you usually specify the user ID in the job definition. You can also define a default user ID in the agentparm.txt file so that all jobs on the agent computer run under the default user ID. If another user ID is specified in the job definition, the agent ignores the default user value and the job runs under the user ID specified in the job definition.

To define a default user ID, configure the following agent parameters on the agent:

oscomponent.default.user

Specifies the default user ID.

oscomponent.default.password

Specifies the password for the default user ID.

Note: This parameter is only required for Windows systems.

How to Set Up Environment Variables

You can define environment variables on the agent that jobs submitted by the agent can access. You can define the following types of environment variables:

- Agent-wide variables that are available for every job from every scheduling manager on behalf of every user.
- Manager-specific variables that are available for every job from a specific scheduling manager on behalf of every user. These variables override agent-wide variables.
- User-specific variables that are available for every job from a specific user. These variables override agent-wide variables and manager-specific variables.

To set up environment variables, do the following:

1. Define environment variables in a file (see page 75).
2. Specify the path to the environment variables (see page 76).

Defining Environment Variables in a File

You define each environment variable on a single line in a text file as a variable=value pair. Create a different file for each type of environment variable. For example, you can create a file named agent_vars.txt to store all your agent-wide environment variables and a file named mgr_stress_vars.txt to store all your manager-wide environment variables.

Notes:

- If the agent is running as root, the agent reads a user-specific variables file using the root authority and does not check permissions of the file for the user specified in a job definition. This rule also applies to manager-specific variables files.
- On Windows, the total environment variable file size must not exceed 16 KB.

Example: Defining Environment Variables

A text file contains the following two manager-specific environment variables required for CA Workload Automation AE.

```
AUTOSYS=C:\Program Files\CA\CA Workload Automation AE\autosys
AUTOROOT=C:\Program Files\CA\CA Workload Automation AE
```

Using the $EWAGLOBALPROFILE environment variable for UNIX Workload

For agents running on UNIX, you can specify the $EWAGLOBALPROFILE environment variable to specify the path to a file or script that defines global variables.

Note: To use the $EWAGLOBALPROFILE environment variable, run the agent as root and set the following parameters in the agentparm.txt file:

- oscomponent.loginshell=true
- oscomponent.lookupcommand=true

Example: Using the $EWAGLOBALPROFILE environment variable

In this example, the $EWAGLOBALPROFILE environment variable is included in a file that defines environment variables. $EWAGLOBALPROFILE specifies the path to the var_ewa_global.txt UNIX script.

```
EWAGLOBALPROFILE=u1/envar/var_ewa_global.txt
```

Specify the Path to the Environment Variables

After you have defined your agent-wide, manager-specific, or user-specific environment variables in separate text files, configure the agent to specify the location of the files.

To specify the path to the text files, configure the following agent parameters on the agent:

Note: If you omit the path, the agent uses the profiles/filename subdirectory of the agent installation directory as the default path.

- To specify the path to agent-wide variables, configure the following parameter:

  oscomponent.environment.variable

  Specifies the path to the file that defines agent-wide variables.

  Example: C:\MyVars\agent_vars.txt

- To specify the path to manager-specific variables, configure the following parameter:

  oscomponent.environment.variable_manager_*managerid*

  Specifies the path to the file that defines manager-specific variables.

  managerid

  Specifies the name of the specific scheduling manager the environment variables apply to.

  Example: C:\MyVars\mgr_stress_vars.txt

- To specify the path to user-specific variables, configure the following parameter:

  oscomponent.environment.variable_user_*userid*

  Specifies the path to the file that defines user-specific variables.

  userid

  Specifies the name of the user the environment variables apply to.

  Example: C:\MyVars\usr_abc.txt

Specify Job Classes and Number of Initiators

The initiator.class parameter sets the maximum number of active jobs of a particular class allowed by the agent. By default, the agent allows up to 1000 jobs of the default class at a given time. You can further control jobs the agent allows by setting up additional initiator classes and indicating which job types they control. For example, you can have a UNIX class that only allows 100 active UNIX jobs at the same time.

To specify job classes and number of initiators, configure the following agent parameters on the agent:

initiators.class_*n*=*jobclass*,*number of initiators*

Describes job classes and the number of initiators that can process jobs that are assigned a particular job class. Use a new line for each initiators.class_*n* parameter, where *n* is an integer starting at the value 1. By controlling the type and number of initiators, you can have greater control over the initiation of jobs and manually balance the loads on system resources.

The parameter initiators.afmjobclassmap_*n* relates to this parameter. However, the value of *n* does not have to match in both parameters.

For UNIX workload, depending on the number of initiators you assign, you may need to increase the number of threads that can be run per process on your operating system.

Examples:

initiators.class_1=Default,1000

initiators.class_2=POJO,100

initiators.afmjobclassmap_*n*=*verb*,*subverb*,*jobclass*

Maps verb and subverb combinations of a job request to a job class. When the agent sees an AFM containing a defined pair of verb and subverb, it assigns the specified job class to that job.

The defined pair must be a valid verb and subverb combination. Write a separate instance of this parameter for each pair. For some job types, you can also specify a job class in a job definition.

Note: To find out which verbs and subverbs you can use, see the agent receiver log.

Example: Set the Job Class and Number of Initiators

Suppose that you want to limit the number of Web Service jobs the agent initiates to 100. In this example, the job class is defined as WS. The following agent receiver log contains the verb and subverb: WEBSERVICE and RUNRPC.

```
AM STRESS HELLO4.TXT/OA.1/MAIN WEBSERVICE RUNRPC TargetNS(http://tempuri.org/)
Operation(HelloWorld)
WSDLURL(http://138.42.98.127:3247/WebSite3/Service.asmx?WSDL)
ServiceName(Service) PortName(ServiceSoap)
```

Note: The verb and subverb in the agent receiver log follow the job ID, HELLO4.TXT/OA.1/MAIN.

To set the job class and number of initiators, you configure the following parameters to the values shown.

```
initiator.class_1=Default,1000
initiator.class_2=WS,100
initiators.afmjobclassmap_1=WEBSERVICE,RUNRPC,WS
```

Configure the Agent to Monitor Available Disk Space

You can configure the agent to monitor the amount of available disk space for the database directory and send notifications to warn you when the space is too low. The agent has three disk space warning thresholds:

- Notice—The agent sends a warning notice when the disk space reaches this level but continues to run.
- Severe—The agent sends a severe warning and stops accepting new automated framework messages (AFMs).
- Critical—The agent logs a critical warning and shuts down.

The agent logs the severe and critical warning messages in the runner_os_component.log and nohup.stderr logs.

To configure the agent to monitor available disk space

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

   ```
   agent.resourcemon.enable=true
   ```

   Note: By default this parameter is set to false.
5. Set one or more of the following parameters:

   agent.resourcemon.threshold.disk.warning.notice

   Specifies the amount of disk space required before the agent sends an SNMP trap. The agent continues to run after it sends the trap. This parameter uses the following syntax:

   `*size*[B|K|M|G]`

   size

   Specifies the amount of disk space in bytes (B), kilobytes (K), megabytes (M), or gigabytes(G).

   Default: 21M

agent.resourcemon.threshold.disk.warning.severe

Specifies the amount of disk space required before the agent sends an SNMP trap and stops accepting new automated framework messages (AFMs). This parameter uses the following syntax:

```
size[B|K|M|G]
```

size

Specifies the amount of disk space in bytes (B), kilobytes (K), megabytes (M), or gigabytes(G).

Default: 20M

Note: The agent resumes accepting new AFMs when the available disk space is greater than the size specified by this parameter.

agent.resourcemon.threshold.disk.critical

Specifies the amount of disk space required before the agent shuts down. This parameter uses the following syntax:

```
size[B|K|M|G]
```

size

Specifies the amount of disk space in bytes (B), kilobytes (K), megabytes (M), or gigabytes(G).

Default: 10M

6. Save and close the agentparm.txt file.
7. Start the agent.

   The agent is configured to monitor available disk space.

Important! To send SNMP traps, you must configure the agent to connect with an SNMP manager.

More information:

Configure the Agent to Connect with an SNMP Manager (see page 102)
SNMP-related Problems (see page 167)

How to Set Up Wake On LAN (WOL)

You can save energy using the Wake on LAN (WOL) feature to automate the startup of your computers. Setting up WOL lets you define and schedule WOL jobs to send a signal to a server to turn it on. When the server is no longer needed, you can schedule a different job to power it down.

Wake on LAN (WOL) is a hardware and software solution that lets you wake up a computer remotely. The solution requires an ACPI-compliant computer and a special software program that sends a signal to the network card of the computer to wake it up. The agent provides the AMD magic packet to broadcast the signal to a computer that has been soft-powered-down (ACPI D3-warm state). You can configure the agent for how many times it broadcasts the signal and the amount of time it waits between broadcasts.

Note: Not all scheduling managers support Wake on LAN. Consult the *Release Notes* for your scheduling manager to determine whether this enhancement is supported.

To set up Wake on LAN (WOL)

1. Collect the MAC address (see page 80).
2. Configure WOL properties on the agent (see page 81).
3. Define a WOL job (see page 82).

Collecting the MAC Address

You require the Media Access Control (MAC) address of the computer you want to receive the Wake on LAN (WOL) signal. The MAC address is burned into the Ethernet card (NIC) of the motherboard.

Configure WOL Properties on the Agent

To run Wake on LAN (WOL) jobs, define specific parameters to enable communication. You can set up how often the agent sends the magic packet to the WOL-enabled computer.

To configure WOL properties on the agent

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

   ```
   plugin.start_internal_N=management
   ```
5. Define the following parameters:

   management.wol.nudges

   Specifies the number of times the agent broadcasts the magic packet.

   Default: 10

   management.wol.nudges.sleep

   Specifies the amount of time, in milliseconds, between broadcasts of the magic packet.

   Default: 1000 (ms)

   management.wol.ports

   Specifies the port that the magic packet is sent to.

   Default: 6
6. Save and close the agentparm.txt file.
7. Start the agent.

   The agent is configured for WOL jobs.

Defining a WOL Job

To define a WOL job, you require the following information:

- The broadcast address where the packet must be broadcasted
- The target computer MAC address. Represented in a '-' or ':' separated list of six octets (bytes) in hexadecimal format.
- The optional IP address to which the agent attempts the connection to verify that the target host is up.
- The optional list of ports to which the agent attempts the connection to verify that the target host is up.

  Defaults: 21, 22, 23, 80, 111, 135, 139, 445
- The space or comma-separated list of ports. Optional. In case none are specified, 0 is assumed.
- The WOL password. Must be 4 or 6 '.', '-', or ':' separated octets (bytes) in hexadecimal format.

For detailed instructions to define a WOL job, see the documentation for your scheduling manager.

Enable Operating System Reporting in the Agent Status

You can enable operating system errors corresponding to the script or binary exit codes sent from the agent to the scheduling manager.

To enable operating system reporting in the agent status, configure the following agent parameter on the agent:

oscomponent.lookuposerror

Enables operating system errors to pass from the agent to the scheduling manager.

- false—Disables the operating system errors.
- true—Enables the operating system errors.

Default: false

Configure the Agent for Windows Interactive Jobs

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

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

To configure the agent for Windows interactive jobs

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

   ```
   oscomponent.interactive=true
   ```

5. Optional. Define the following parameter:

   oscomponent.shell

   Specifies the executable file name for an alternative Windows shell.

   Default: explorer.exe

6. Save and close the agentparm.txt file.
7. Start the agent.

   The agent is configured for Windows interactive jobs.

More information:

Problems with Windows Interactive Jobs (see page 168)

How to Configure the Agent to Run as a Windows Service

You usually run the agent on Windows as a background application by running it as a Windows service. However, you can run the agent in console mode to diagnose a problem. Running the agent as a Windows service lets you access remote resources by running a job under a user ID. To run jobs under different user IDs, configure the agent to run as a Windows service under the local system account. Although not recommended, you can configure the agent to run as a Windows service under the local user account if you only run jobs under one user ID.

To configure the agent to run as a Windows service, follow these steps:

1. Register the agent as a Windows service (see page 84).
2. Configure the agent service to run under the local system account option (see page 85).

   The local system account is the default set when the agent is registered as a Windows service.

Register the Agent as a Windows Service

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

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

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

To register the agent as a Windows service

1. Click the Install Agent Windows Service program shortcut.
2. From the command prompt, change to the agent installation directory, and issue the following command:

   ```
   cybAgent -install
   ```

Deregister the Agent as a Windows Service

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

To deregister the agent as a Windows service

1. Click the Remove Agent Windows Service program shortcut.
2. From a command prompt, change to the agent installation directory, and issue the following command:

   ```
   cybAgent -remove
   ```

Configure the Agent Service to Run Under the Local System Account

When you register the agent as a Windows service, the service is configured to run under the local system account by default. If you must reconfigure the agent service to run under the local system account, follow these steps:

1. Stop the agent.
2. Open the Windows Control Panel.
3. Double-click Administrative Tools.
4. Double-click Services.
5. Right-click the agent service and click Stop.
6. Right-click the agent service and select Properties.
7. On the Properties dialog, select the Log On tab.
8. Select Local System account.
9. Click OK.

When running Windows programs as a service, you are restricted to how you can access data on remote computers.

Under a system account, note the following:

- No user is running the process and, therefore, the service has limited access to network resources, such as shared directories and pipes.
- Services use null session support to interact with the desktop and can connect to resources using a null session.

When a program is started using the system account, it logs on with null credentials. If it tries to access a remote resource using a null session, it fails. To avoid this problem, specify the user ID under which resources are accessed in your job definition. Using this approach, you do not have to modify the Registry.

You can also have the system administrator change Registry values, however, we do not recommend this practice. Using the Registry Editor can cause serious problems, which can require reinstallation of the Windows operating system. If you change the Registry values, do one of the following:

- Set the value of the RestrictNullSessAccess parameter to FALSE (the default value is TRUE).
- Specify lists of share names and pipe names accessed by the system account, using the NullSessionShares and NullSessionPipes parameters respectively. These names must be specified on the computers where the resources exist. By default, the only share names the agent service can access are those names listed with the NullSessionShares parameter.

Note: Verify that you are complying with the terms of the agent license agreement before accessing network resources with the agent. In most situations, you are permitted to access data on remote computers. Scripts or executable files run by the agent, however, must use the CPU and memory of the computer where the agent resides.

Configure the Agent Service to Run Under the Local User

Although not recommended, you can also run the agent from a local user account. When you start the service under the This account option, it runs using the security context of the specified user account. If the user account and password are valid, the service process has access to network resources.

Note: When the agent service is configured to run under a local user account, you cannot easily run jobs using different user IDs, and certain programs cannot run. When you access a remote computer using the agent on Windows, the user ID defined in the job statement or in the This account option is a domain user. If the local and remote servers are standalone servers, you must have the same user IDs and passwords defined on both servers.

To configure the password for the specified user account

1. Stop the agent.
2. Open the Windows Control Panel.
3. Double-click Administrative Tools.
4. Double-click Services.
5. Right-click the agent service and click Stop.
6. Right-click the agent service and select Properties.
7. On the Properties dialog, select the Log On tab.
8. Select This account.

   Local System is the default entry in the first field.
9. Type the password in Password.
10. Retype the password in Confirm password.
11. Click OK.

Set PAM Parameters for User Authentication on UNIX Systems

PAM (Pluggable Authentication Modules) is used for security to check whether a service should be used. A service is a program that provides a function that requires authentication. Examples of services are login, sshd, pam, and sudo.

To set PAM parameters for user verification on UNIX systems, configure the following agent parameters on the agent:

oscomponent.auth.pam.svc

Specifies the default PAM service the agent uses for login authentication. The list of available PAM services for your system is located in the /etc/pam.conf or /etc/pam.d/ file.

Default: login

Note: You can use the chkusr utility provided with the agent to test a PAM service being used to authenticate a user and password.

oscomponent.auth.pam.lib

Optional. Specifies the path to the PAM shared library.

Note: We recommend that you specify the full path to the library file.

Test User Authentication Settings

The chkusr utility, provided with the agent, lets you test the authentication settings on your system for a user. For example, you can test a PAM service on your system to see whether it can be used to authenticate a user.

To test user authentication settings

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

   ```
   $ ./chkusr user password [pam_service]
   ```

   user

   Specifies the user that requires authentication.

   password

   Specifies the encrypted password corresponding to the user.

   Note: Use the password utility to encrypt the password

   pam_service

   Optional. Specifies the PAM service that authenticates the user.

   The chkusr utility displays a validation message.

Example: Testing user authentication settings

The following command tests the sshd PAM service for the osuser.

```
$ ./chkusr osuser FD5AD34EC7A8F07C0B2BE8 sshd
```

More information:

Encrypt a Password Using the Password Utility (see page 120)

Force the Default Shell for UNIX Jobs

The shell the agent uses to run a UNIX script is determined by the following settings in the order listed:

- The shell specified in the job definition.
- The first line of the script the job runs.
- The oscomponent.defaultshell parameter in the agentparm.txt file for the agent.
- The user default shell defined in the user profile.

You can force the agent to use the default shell defined by the oscomponent.defaultshell parameter in the agentparm.txt file.

To force the default shell for UNIX jobs

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

   ```
   oscomponent.defaultshell.force=true
   ```
5. Save and close the agentparm.txt file.
6. Start the agent.

   The agent uses the default shell to run UNIX scripts and to source the user's profile.

Configure the Agent for Monitoring Jobs

Depending on the type of file activity the agent monitors, you can configure the following parameters, to skip the first scan of a monitored file:

filemon.firstscan.skip

Sets whether the agent skips the first scan of a monitored file for CREATE, UPDATE, SHRINK, or EXPAND file activity.

- false—Uses the first, as well as subsequent scans, for the file system activity monitoring.
- true—Skips the first scan of the monitored file.

Default: false

Note: You can set this parameter to true for backward compatability with AutoSys agents.

filemon.update.firstscan.skip

Sets whether the agent skips the first scan of a monitored file for UPDATE activity with no change.

- false—Uses the first, as well as subsequent scans, for the file system activity monitoring.
- true—Skips the first scan on update of the monitored file.

Default: false

Note: This parameter applies to File Trigger jobs, and when set to true, emulates R7 agent behavior.

Configure the Agent to Run File Triggers as Separate Processes

You can configure the agent to run file triggers as separate processes, which lets the agent do the following:

- Scan files using a specified user ID on a UNIX or Linux system, letting the agent scan NFS file systems that the local root user does not have access to.
- Resolve file names to be scanned on both UNIX and Windows.

To configure the agent to run file triggers as separate processes

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

   ```
   filemonplugin.runexternal=true
   ```

   This setting lets the agent start as an external process and communicate using Named Pipes (Windows) or SysV IPC (UNIX/Linux) queues.
5. Specify the following parameter to set a default user for submitting jobs:

   oscomponent.default.user

   Specifies the default operating system user name.

   Note: The user specified in a job definition overrides this value.
6. (Windows only) Specify the following parameter:

   oscomponent.default.password

   Specifies the corresponding Windows password for the default user.
7. Save and close the agentparm.txt file.
8. Start the agent.

   The agent submits each file trigger as a separate process.

Configure the Agent for auto_remote

Note: This procedure applies to CA Workload Automation AE only. The r11.3 documentation refers to auto_remote as the legacy agent.

CA Workload Automation Agent for UNIX, Linux, or Windows replaces the Remote Agent (auto_remote) that was provided with Unicenter AutoSys JM r4.5 and r11. By default, the r11.3 agent acts differently than auto_remote for sourcing job profiles and global profiles from within a script when a user's profile sourcing is disabled. You can configure agent parameters so that the r11.3 agent acts similar to auto_remote.

To configure the agent for auto_remote

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

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

   The agent will execute a temporary shell script without sourcing the user's profile.
5. Set the following parameter:

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

   The agent drops the sourcing of the profiles into the temporary shell script.
6. Set the following parameter:

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

   - If the job profile is present, the agent sources the job profile. The EWAGLOBALPROFILE (/etc/auto.profile) is not sourced.
   - If the job profile is not present, the agent sources EWAGLOBALPROFILE.
7. Save and close the agentparm.txt file.
8. Start the agent.

Chapter 6: Configuring the Agent as an SNMP Manager

This section contains the following topics:

Configure the Agent as an SNMP Manager

You can configure an SNMP agent plug-in, packaged with the agent, to act as an SNMP manager to emit and listen for SNMP traps. The SNMP agent plug-in supports SNMP V1, V2, and V3. Once configured, users can define and run SNMP job types.

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

To configure the agent as an SNMP manager, configure the following agent parameters on the agent:

snmp.response.translate

Sets whether the agent translates Object Identifiers (OIDs).

- false—Disables translation.
- true—Enables translation.

Default: false

snmp.response.translate.full

Sets whether the agent translates the Object Identifiers (OIDs) from the numeric format to the string format.

- false—Disables full-name translation.
- true—Enables full-name translation.

Default: false

snmp.request.timeout

Defines the time-out, in milliseconds (ms), when the agent requests SNMP trap information.

Default: 2000 (ms)

snmp.request.retries

Defines the maximum number of times the agent requests SNMP trap information. Zero indicates one attempt.

Default: 0

Configure the SNMP Trap Listener for SNMP Subscribe Jobs

To configure the SNMP trap listener, configure the following agent parameters on the agent:

snmp.trap.listener.version

Specifies the SNMP version of the SNMP manager you want the agent to connect with.

- 1—Specifies SNMP v1.
- 2—Specifies SNMP v2.
- 3—Specifies SNMP v3.

Default: 2

snmp.trap.listener.host

Specifies the IP address of the agent listening for trap information.

snmp.trap.listener.port

Specifies the agent port listening for trap information.

Default: 162

Limits: 1-65535

snmp.trap.listener.community

Specifies the v1 or v2 SNMP trap community. The SNMP trap listener ignores traps that do not match this community type.

Default: public

snmp.trap.listener.v3.auth.password_*n*

Specifies the encrypted authentication password for the SNMP v3 user, where *n* is an integer starting from 1.

Note: All parameters ending with the same value of n belong to the same group. This parameter applies only to SNMP v3.

snmp.trap.listener.v3.auth.protocol_*n*

Specifies the authentication protocol of the SNMP trap listener, where *n* is an integer starting from 1.

- MD5—Specifies the Message Digest 5 Algorithm.
- SHA—Specifies the Secure Hash Algorithm.

Note: All parameters ending with the same value of n belong to the same group. This parameter applies only to SNMP v3.

snmp.trap.listener.v3.engine_*n*

Specifies the agent engine ID that sends trap information, where *n* is an integer starting from 1. All parameters ending with the same value of n belong to the same group. This parameter applies only to SNMP v3.

Default: AGENT_ENGINE

snmp.trap.listener.v3.priv.password_*n*

Specifies the encrypted privacy password for the SNMP v3 user, where *n* is an integer starting from 1.

Note: All parameters ending with the same value of n belong to the same group. This parameter applies only to SNMP v3.

snmp.trap.listener.v3.priv.protocol_*n*

Specifies the privacy protocol for the SNMP v3 user, where *n* is an integer starting from 1.

- AES—Specifies the Advanced Encryption Standard that uses a 32-character encryption key. AES is the algorithm required by U.S. Government organizations to protect sensitive (unclassified) information (FIPS-140-2 compliance).
- DES—Specifies the Data Encryption Standard that uses a 16-character encryption key.

Note: All parameters ending with the same value of n belong to the same group. This parameter applies only to SNMP v3.

snmp.trap.listener.v3.user_*n*

Specifies the user authorized to communicate with the SNMP v3 agent, where *n* is an integer starting from 1.

Note: All parameters ending with the same value of n belong to the same group. This parameter applies only to SNMP v3.

Filter Interpretation for SNMP Subscribe Jobs

An SNMP Subscribe job uses regular expressions to filter SNMP trap information. The agent checks the job filter against an OID(Value) string that is composed of SNMP VarBinds. The order of the SNMP VarBinds is dictated by the SNMP trap sender. For SNMP version 1 only, the agent inserts a trap timestamp as the first VarBind.

By default, the agent does not attempt translation of the Object Identifiers (OIDs) from a numeric format to a string format. You can change the translation by setting the following parameters to true:

- snmp.response.translate
- snmp.response.translate.full

The agent attempts the translation prior to checking an SNMP Subscribe job filter against the OIDs. If the translation fails, the agent uses the OIDs in numeric format.

The agent logs the information sent in the SNMP VarBinds in the plugin_log_snmp.log and is prefixed with "Matching string for *WOB ID*".

Chapter 7: Configuring Agent Aliases for Clustered Environments

This section contains the following topics:

How to Configure Agent Aliases for Clustered Environments

If a node fails or is down for maintenance, cluster management software migrates application packages from the inactive node to an active node in the cluster. The agent aliases let the agent accept and respond to automated framework messages (AFMs) for migrated packages. The packages are necessary to continue workload processing when the node where the workload was running experiences failover.

The scheduling manager administrator configures each application package as an alias agent in the server. This configuration lets the scheduling manager redirect AFMs to the appropriate node where the package is currently running. If the agent on the node the package was migrated to is aware of the package through the alias, when failover occurs, the agent can respond to AFMs for the migrated package.

Although the agent responds to AFMs for one of its aliases, it does not respond to the scheduling manager as the agent on whose behalf the agent responded. In all communications with the scheduling manager, the agent correctly identifies itself using the agentname parameter in the agentparm.txt file.

You can use the agent aliases in clustered environments, such as HACMP/6000 for IBM AIX, MC/ServiceGuard on HP-UX and Linux, and VERITAS Cluster Service on Windows 2000.

To configure agent aliases for clustered environments, follow these steps:

1. Enable aliasing on the agent (see page 98).
2. Enable the agent aliasing on the scheduling manager (see page 98).

Enable Aliasing on the Agent

Agent aliasing is part of the process you can use to set up failover for the agent.

To enable aliasing on the agent, do the following:

1. Install the agent on a partition mounted locally on each node of the cluster.
2. Verify that each agent has its own copy of the agentparm.txt file, log directory, and log files. These files must reside on a locally mounted partition.
3. Configure the spooldir parameter in the agentparm.txt file to help ensure that the agents of the cluster share a common spool directory.

   Note: Sharing a common spool directory ensures that, when failover occurs and a workload object restarts on another node, the agent can retrieve the spool file and continue updating it as required.
4. Configure the communication.alias_n parameter on the agent of each node to enable each agent to respond to AFMs for all alias agents in the cluster.

Example: Configure Agent Aliases for a Clustered Environment

The following example shows how to configure the agentparm.txt file in a two-node clustered environment with two application packages.

Parameter	Node A	Node B
Agentname	AGENTA	AGENTB
Communication.alias_1	PKG1	PKG1
Communication.alsias_2	PKG2	PKG2
Persistence.coldstart	FALSE	FALSE
Spooldir	shareddisk/dir/spool	shareddisk/dir/spool

Enable the Agent Aliasing on the Scheduling Manager

Once you set up aliasing on the agent, enable the agent aliasing on the scheduling manager.

To enable the agent aliasing on the scheduling manager

1. Configure each application package as an alias agent in addition to configuring each agent on the scheduling manager.
2. Select Keep alive only for the physical agent, not for the alias agents you configured for application packages.

Considerations for Alias-Enabled Agents in Clustered Environments

When working with alias-enabled agents in clustered environments, consider the following:

- If you shut down the agent in an alias-enabled clustered environment for maintenance, the agent resumes monitoring as usual when you restart it.
- In job definitions, schedulers must refer to aliased agents, not physical agents.
- To avoid error messages when you restart an alias-enabled agent in a clustered environment, always use a cold start.

Chapter 8: Connecting the Agent to External Applications

The agent has built-in management connectors that let third-party tools monitor and control the agent.

This section contains the following topics:

Configure the Agent to Connect with a JMX Console

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

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

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

management.connector_*n*=jmx

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

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

management.jmx.host

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

management.jmx.port

Specifies the port where the JMX connector listens.

Default: 1099

Configure the Agent to Connect with an SNMP Manager

An SNMP connector, built into the agent, lets you use an SNMP manager to monitor and control the agent. You can use any SNMP manager that supports SNMP v1, v2, or v3 to perform the following tasks on the agent:

- Discover metrics
- Query and modify values of various metrics
- Subscribe and receive notifications through SNMP traps

The cybermation.mib file, located in the agent installation directory, is a Management Information Base (MIB) file that describes all the metrics and SNMP traps for the agent.

To configure the agent to connect to an SNMP manager, configure the following agent parameters on the agent:

management.connector_*n*=snmp

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

Specify snmp to use an SNMP manager to monitor and control the agent.

management.snmp.agent.version

Specifies the SNMP version of the SNMP manager you want the agent to connect with.

- 1—Specifies SNMP v1.
- 2—Specifies SNMP v2.
- 3—Specifies SNMP v3.

Default: 2

management.snmp.mibfile

Specifies the path to the MIB file that describes the metrics and SNMP traps for the agent.

Default: *agentinstalldir*/cybermation.mib

management.snmp.host

Identifies the SNMP Manager IP address or DNS name. Your SNMP administrator can provide the host name.

management.snmp.port

Specifies the SNMP Manager UDP port. Your SNMP administrator can provide this port number.

Default: 162

Limits: 1-65535

management.snmp.community

Specifies the type of network the SNMP traps are sent across for SNMP v1 or v2 only. Your SNMP administrator can provide the type.

- public—Identifies an unsecured network, for example, the Internet.
- private—Identifies a secure network, for example, a local area network.

Default: public

management.snmp.agent.community.read

Specifies the SNMP read community. This parameter applies only to SNMP v1 and v2.

- public—Specifies read-only access.
- private—Specifies read/write access.

management.snmp.agent.community.write

Specifies the SNMP write community. This parameter applies only to SNMP v1 and v2.

- public—Specifies read-only access.
- private—Specifies read/write access.

management.snmp.agent.trapsink.host

Specifies the host name or the IP address of the SNMP listener that receives trap information. The management connector uses this host to send the trap.

management.snmp.agent.trapsink.port

Specifies the port of the SNMP listener that receives trap information. The management connector uses this port to send the trap.

Default: 162

management.snmp.agent.trapsink.community

Specifies the SNMP community that receives trap information.

- public—Specifies read-only access.
- private—Specifies read/write access.

Default: public

management.snmp.agent.trapsink.user

Specifies the user authorized to receive trap information.

Configure Connection with a Version 3 SNMP Manager

To configure the agent to connect with a Version 3 SNMP Manager, configure the following agent parameters on the agent:

management.snmp.agent.user

Specifies the user authorized to communicate with the SNMP agent plug-in.

Example: MBAGENT

management.snmp.agent.user.auth.protocol

Specifies the authentication protocol the agent uses. Supported protocols are SHA and MD5.

Example: SHA

management.snmp.agent.user.auth.password

Specifies the encrypted authentication password for the user authorized to communicate with the SNMP agent plug-in.

management.snmp.agent.user.priv.protocol

Specifies the privacy protocol the agent uses. Supported protocols are AES and DES.

Example: AES

management.snmp.agent.user.priv.password

Specifies the encrypted privacy password for the user authorized to communicate with the SNMP agent plug-in.

Chapter 9: Setting Up Security

This section contains the following topics:

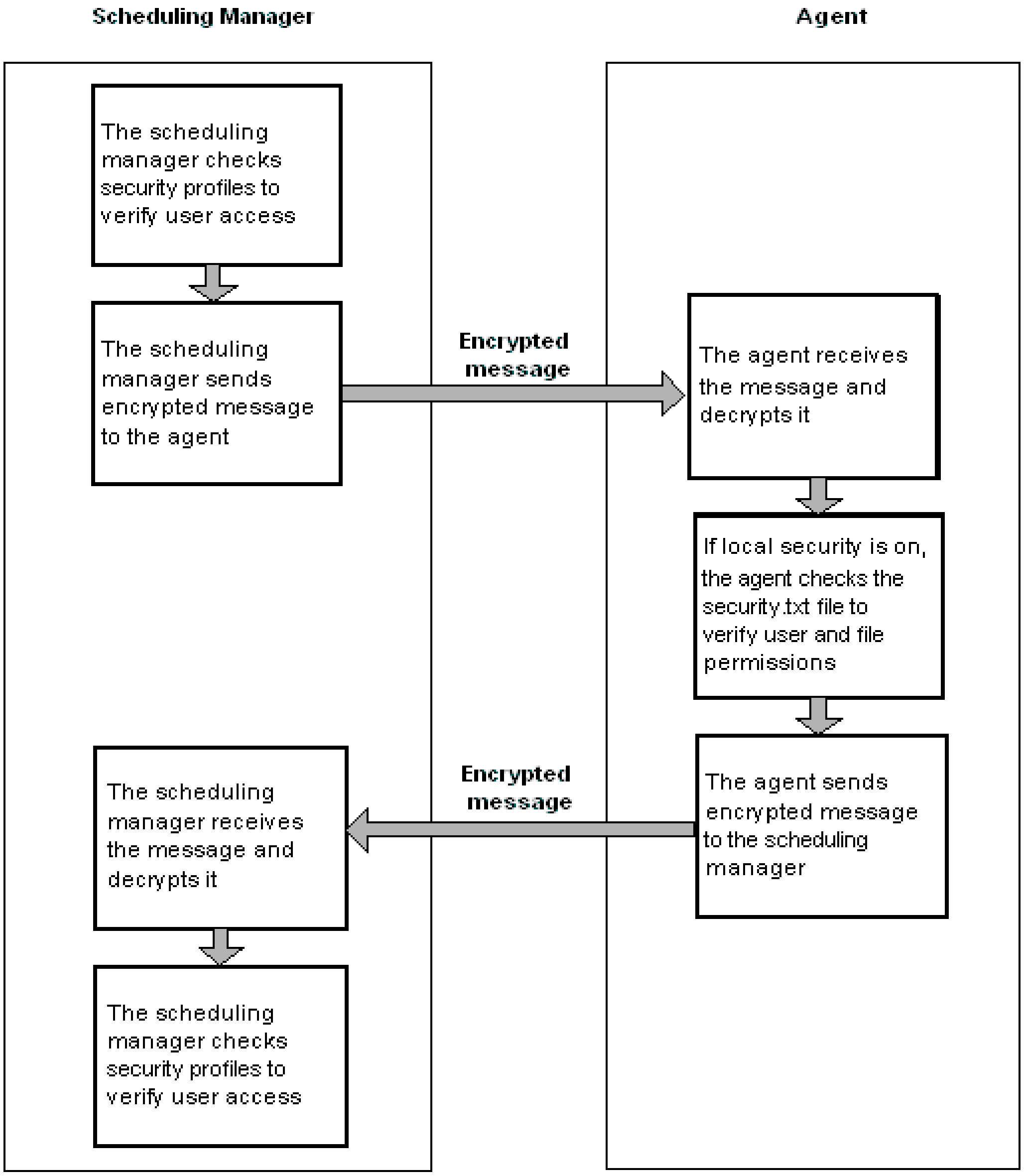
Types of Security

At a minimum, security is set between the agent and the scheduling manager using an encryption key. The agent requires encrypted communication with the scheduling manager. The encryption key is set when you install the agent and when you configure the scheduling manager to work with the agent.

You can also set up local security on the agent to control the following:

- Which scheduling manager user IDs can submit jobs under a specific agent user ID, from a specific directory
- Which FTP user IDs can issue FTP-related commands to files in directories
- Which scheduling manager user IDs can issue control commands and send messages to an agent

Example: Security between the agent and the scheduling manager


Scheduling Manager
Agent
The scheduling manager checks security profiles to verify user access
The scheduling manager sends encrypted message to the agent
Encrypted message
The agent receives the message and decrypts it
If local security is on, the agent checks the security.txt file to verify user and file permissions
The agent sends encrypted message to the scheduling manager
Encrypted message
The scheduling manager receives the message and decrypts it
The scheduling manager checks security profiles to verify user access

The scheduling manager verifies the access privileges of the user it has even before sending workload to the agent. The scheduling manager also verifies security when receiving messages from the agent. The agent also has its own security verifications it performs when it receives instructions from the scheduling manager.

How to Set Up Security between the Agent and the Scheduling Manager

Encryption is a mandatory security feature that safeguards communication between the agent and the scheduling manager. Your scheduling manager administrator must complete configuration tasks so that the agent and the scheduling manager can communicate with message encryption.

To set up security between the agent and the scheduling manager, follow these steps:

1. Set up security permissions on the scheduling manager (see page 107).
2. Set the encryption on the agent (see page 108).
3. Set the encryption key on the scheduling manager (see page 110).
4. Restart the agent (see page 110).
5. Run a test job to test the security.

   For detailed instructions to run a test job, see the documentation for your scheduling manager.

Security Permissions on the Scheduling Manager

Your scheduling manager administrator must set up the following security permissions on the scheduling manager to control agent access:

- Permission to run work on the agent
- Permission to run a job on the agent under a user ID
- Permission for the agent to issue control commands

Note: For more information about security permissions, see the documentation for your scheduling manager.

Set the Encryption on the Agent Using the Keygen Utility

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

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

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

To set the encryption on the agent using the keygen utility

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

   ```
   keygen 0xkey cipher destination
   ```

   key

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

   - AES—32 hexadecimal character encryption key.

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

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

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

   Note: Not all scheduling managers support all the encryption types. Consult the documentation for your scheduling manager to determine which encryption types are supported.

cipher

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

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

Default: DES

destination

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

Default: cryptkey.txt

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

The keygen utility replaces the encryption key.

Example: Encrypt a Key

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

```
keygen 0x1020304050607080 DES
```

Configure the Agent for No Encryption

You can configure the agent for no encryption.

To configure the agent for no encryption

1. Change to the agent installation directory.
2. Stop the agent.
3. Open the agentparm.txt file located in the agent installation directory.
4. Set the security.cryptkey parameter to no value, as follows:

   ```
   security.cryptkey=
   ```
5. Save and close the agentparm.txt file.
6. Start the agent.

   The encryption is disabled on the agent.

Set the Encryption Key on the Scheduling Manager

The scheduling manager and the agent must have the same encryption key to communicate. The encryption key for the agent is stored in a text file. The security.cyrptkey parameter in the agentparm.txt file sets the path to the text file. After you set the encryption key on the agent, set the same key on the scheduling manager. If the keys are different, the agent and scheduling manager cannot communicate and an AGENTDOWN state occurs when you try to run workload.

Note: For detailed instructions to set the encryption key on the scheduling manager, see the documentation for your scheduling manager.

Restart the Agent

After you have set up encryption on the agent, restart the agent to complete the configuration.

To restart the agent

1. Ensure that you are in the agent installation directory.
2. Stop the agent using one of the following commands:
   - On UNIX:

     ```
     ./cybAgent -s
     ```

   - On Windows:

     ```
     cybAgent -s
     ```

   The agent stops running.
3. Start the agent using one of the following commands:
   - On UNIX:

     ```
     ./cybAgent &
     ```

   - On Windows:

     ```
     cybAgent -a
     ```

   The agent restarts.

Configure the Agent for Encryption Standard FIPS 140-2

The U.S. Government encryption standard FIPS 140-2 requires a FIPS-certified library and FIPS-certified cipher algorithm. To comply with the standard, the agent provides the following:

- RSA BSAFE Crypto-J library
- Advanced Encryption Standard (AES) cipher algorithm

If you did not select the AES cipher algorithm when you installed the agent, you can configure the agent to comply with encryption standard FIPS 140-2.

To configure the agent for encryption standard FIPS 140-2

1. Change to the agent installation directory.
2. Stop the agent.
3. Open the agentparm.txt file.
4. Edit the following parameter to specify the encryption key:

   security.cryptkey

   > Defines the encryption key the agent uses to communicate with the scheduling manager. The encryption key must be prefixed with 0x and followed by the number of characters required for the chosen cipher algorithm:
   >
   > - AES—32-character encryption key
   > - DESEDE—48-hexadecimal encryption key
   >
   > **Note:** Encrypt the encryption key using the keygen utility.

5. Set the following parameter for the agent to use the FIPS-certified library and cipher algorithm.

   ```
   security.jce.fips=true
   ```

   Note: Setting this parameter can impact workload that uses SSL, for example, FTP jobs where the servers do not use the same cipher suites.

6. Save and close the agentparm.txt file.
7. Start the agent.

More information:

Configure Agent Parameters on the Agent (see page 56)

How to Set Up Local Security on the Agent

The agent has its own security verification it performs when it receives instructions from the scheduling manager. Security rules on the agent define the local security verification. Local security on the agent controls which scheduling manager user IDs can perform the following actions:

- Submit jobs run under a specific agent user ID.
- Issue CONTROL messages to the agent.
- Perform FTP transfers under a specific agent user ID.

Note: Agent security rules do not override permissions set at the operating system level.

To set up local security on the agent, follow these steps:

1. Enable local security (see page 112).
2. Configure the security.txt file (see page 113).
3. Refresh the security.txt file (see page 119).

Enable Local Security

Local security must be enabled for the agent to perform its own security checks.

To enable local security

1. Stop the agent.
2. Open the agentparm.txt file located in the agent installation directory.
3. Define the following parameter:

   ```
   security.level=on
   ```
4. Save and close the file.
5. Start the agent.

   Local security is enabled on the agent.

Configure the security.txt File

The security.txt file contains the rules that allow or deny the scheduling manager user IDs the authority to issue control commands to the agent.

Note: If the security.txt file does not exist, default security rules apply.

To configure the security.txt file

1. Open the security.txt file, or create one if it does not exist.

   Note: The security.txt file must reside in the agent installation directory.

2. Define security file rules in the security.txt file (see page 114).
3. Save and close the file.

Example: Security File Rule

The following line in the security.txt file allows all users to issue all control commands to the agent.

```
c a * CONTROL *
```

Security File Rules

The security file contains the following three types of rules:

```
x a | d manager_userID agent_userID path
```

Defines a rule that allows or denies the scheduling manager user IDs from submitting jobs that run under a specific user ID, from a specific directory. These rules begin with the letter x.

x

Identifies a rule controlling execution of scripts and commands.

a | d

Specifies whether access is allowed or denied.

- a—Indicates permission is allowed.
- d—Indicates permission is denied.

manager_userID

Defines the scheduling manager name or the scheduling manager user ID this rule applies to.

agent_userID

Defines the user ID on the agent computer under which the job runs.

path

Defines the path that the scheduling manager is allowed to submit jobs from, using the user ID identified by *agent_userID*. Paths are case sensitive.

Note: On CA Workload Automation AE, jobs are always submitted to run under the user specified in the owner attribute. If local security is enabled on the agent, the agent verifies the permissions of the job owner only. The agent does *not* verify the CA Workload Automation AE user who submits the job. For CA Workload Automation AE, you can define the security rule as follows:

```
x a | d job_owner agent_userID path
```

job_owner

Defines the user specified in the owner attribute.

```
f a | d FTP_userID operation path
```

Defines a rule that allows or denies FTP user IDs from issuing FTP-related commands to files in specified directories. These rules begin with the letter f.

f

Identifies FTP commands.

a | d

Specifies whether access is allowed or denied.

- a—Indicates permission is allowed.
- d—Indicates permission is denied.

FTP_userID

Defines the FTP user ID this rule applies to.

operation

Specifies the FTP command. Valid commands are as follows:

- list—Changes directory and list files (CD, LIST, NLST)
- read—Retrieves the file (RETR)
- write—Stores the file or makes a directory (STOR, STOU, RNFR, RNTO, MKD)
- delete—Deletes the file or directory (DELE, RMD)

These commands apply to the agent as FTP server. For FTP jobs, only read and write commands apply.

path

Specifies the path that the scheduling manager is allowed to submit jobs from, using the user ID identified by Agent_UserID. Paths are case sensitive.

```
c a | d manager_userID CONTROL command
```

Defines a rule that allows or denies scheduling manager user IDs the authority to issue control commands to the agent. These rules begin with the letter c.

Note: This rule does not apply to CA Workload Automation AE.

c

Identifies a rule controlling operational commands to an agent.

a | d

Specifies whether access is allowed or denied.

- a—Indicates permission is allowed.
- d—Indicates permission is denied.

manager_userID

Defines the scheduling manager name or the scheduling manager user ID this rule applies to.

command

Specifies the control command. Valid commands are shutdown, refresh, flush, quiesce, and restart. You can also specify an asterisk (*) for all commands.

Notes:

- Specify at least one rule of each type (x, f, and c) in the security.txt file.
- If security.txt does not exist, default security rules apply.
- Agent security rules do not override permissions set at the operating system level.
- To specify an f rule that restricts access to a directory itself (not the contents in the directory), the directory path must end with a forward slash.

Example: Scheduling Manager (x) Security File Rules

The following rule allows any scheduling manager user to submit jobs that use any files from any directory. The user can submit the jobs under any user ID on the agent computer.

```
x a * * +
```

The following rule denies any scheduling manager user from submitting jobs under gem, or any user IDs that begin with gem, from all directories.

```
x d * gem* +
```

The following rule allows any scheduling manager user to submit jobs as root that are named employee, or begin with employee, from the /prod/ directory.

```
x a * root /prod/employee+
```

The following rule allows any scheduling manager user to submit jobs that use any object whose name starts with F in the USER library. The user can submit the jobs under any user profile that starts with the characters USR on the agent computer.

```
x a * USR* /QSYS.LIB/USER.LIB/F*
```

The following rule allows any scheduling manager user to submit jobs that use any object whose name starts with F in the USER library. The user can submit the jobs under any user profile that starts with the characters JO on the agent computer. Members of file objects whose names start with F are included.

```
x a * JO* /QSYS.LIB/USER.LIB/F*
```

The following rule denies any scheduling manager user from submitting jobs that use /QSYS.LIB/MLIB.LIB/DEPT.FILE/PAYROLL.MBR on the agent computer.

```
x d * * /QSYS.LIB/MLIB.LIB/DEPT.FILE/PAYROLL.MBR
```

Example: FTP (f) Security File Rules

The following rule denies all users from using any FTP operations in any directory. To allow specific FTP access, the FTP rules that follow override this general rule.

```
f d * * +
```

The following rule allows all users to list the files in /pub/ftp and its subdirectories:

```
f a * list /pub/ftp/+
```

The following rule allows all users to store files, rename files, and make directories in /pub/ftp/upload and its subdirectories:

```
f a * write /pub/ftp/upload/+
```

The following rule allows all users to read files from /pub/ftp/download and its subdirectories:

```
f a * read /pub/ftp/download/+
```

Example: Command (c) Security File Rule

The following rule allows all users to issue control commands to the agent.

```
c a * * *
```

Additional Formats for Security File Rules

When defining security rules in the security.txt file, you can use the following additional formats:

Wildcards

The scheduling manager name, user IDs, object names, paths, verbs, and subverbs can contain a single wildcard character at the end of the value only.

The following wildcards are valid:

- Asterisk (*)—Represents zero or more character matches in the current directory only.
- Plus sign (+)—Represents zero or more character matches in the current directory and all subdirectories. For a FILE object, + applies to the members within it.

Start point and spacing

Every security rule starts in column 1. Separate items on a line by one or more blanks or tab characters, and end with a new-line character.

Comment lines

The file can contain comment lines. An asterisk (*) or a number sign (#) in column 1 identifies comment lines.

Security Rule Interpretation

For a rule to match, three components of a rule have to match. If two or more rules match, the closest match overrides the others, as follows.

Interpretation	Explanation
A specific rule overrides a generic rule. A generic rule is a rule that contains wildcards.	/u1/jsmith overrides /u1/jsmith* CYBDL01 overrides *CYB
If both rules are generic, the more specific one overrides the other.	/u1/jsmith/scripts/* overrides /u1/jsmith* /u1/jsmith/scripts/a* overrides /u1/jsmith/scripts*
If there is still ambiguity after these rules have been applied, a deny rule overrides an allow rule.	c d USR* * *overrides c a USR* * *

How Local Security Works

This section describes how the agent determines what to validate when it receives instructions from a scheduling manager. By understanding how local security works, you can decide how to configure local security for your system.

When the agent starts, it verifies local security and does the following:

- If local security is enabled, the agent then looks for the security.txt file.
  - If the security.txt file does not exist, default security rules apply.
  - If the security.txt file exists, the agent uses the rules defined in the file. The agent does not use the default security rules. If a request does not have a match in the security file, the agent denies the request.
- If local security is not enabled, the agent does not verify security.

Default Security Rules

When the agent starts, it looks to see whether local security is turned on. If local security is turned on (security.level is set to on in the agentparm.txt file), the agent then looks for the security file.

After installation, the security.txt file contains the following security rules:

```
c a * * *
f d * * +
x d * * +
```

The following default security rules apply when the security file does not exist and agent security is turned on in the agentparm.txt file (security.level=on):

```
x a * * +
x d * root +
c a * * *
f a * * +
```

Note: For CA WA Agent for Windows, substitute administrator for root.

Refresh an Agent Security File

Refresh the agent security.txt file for any changes to take effect.

To refresh an agent security file

From the command prompt, enter the following command:

```
cybAgent -r.
```

Test the Encryption between the Agent and the Scheduling Manager

To test the encryption between the agent and the scheduling manager, run and monitor a job.

Note: For more information about defining jobs, see the documentation for the scheduling manager.

Encrypting and Changing Passwords

When you change a password, encrypt it before entering it in the agentparm.txt file. To encrypt a password, use the Password utility provided with the agent.

The agent handles FTP user IDs and passwords separately.

Encrypt a Password Using the Password Utility

When you change a password, encrypt it for inclusion in the agentparm.txt file. To encrypt a password, use the Password utility provided with the agent.

To encrypt a password using the password utility

1. Change to the agent installation directory, and enter the following command:

   - On UNIX systems, type **password.sh** at the command prompt.
   - On Windows systems, type **password.bat** at the command prompt.

   A command prompt appears asking you to enter your password.

2. Enter your password.

   The utility responds with your encrypted password as in the following example:

   ```
   **** PASSWORD ENCRYPTION ****
   Enter your password: Lisa
   Encrypted password: 1FF8897FCDDF8A62
   ```

3. Exit the utility.

Chapter 10: Setting Up and Running FTP Workload

This section contains the following topics:

FTP Client and FTP Server

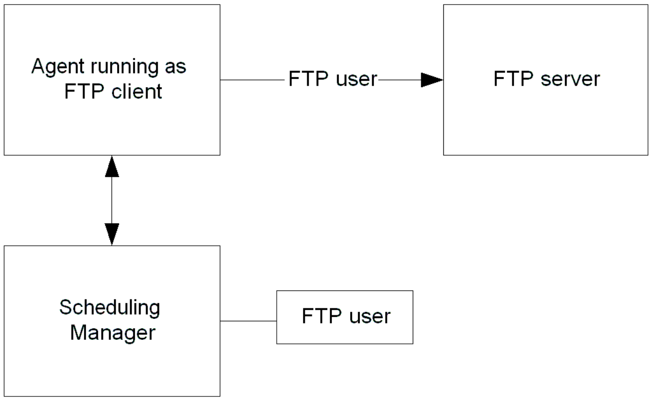
Using your agent, you can automate FTP transfers with FTP jobs. An FTP job can use an existing FTP server or the agent as an FTP server. The FTP job always acts as an FTP client.

Use an FTP job to automate the following:

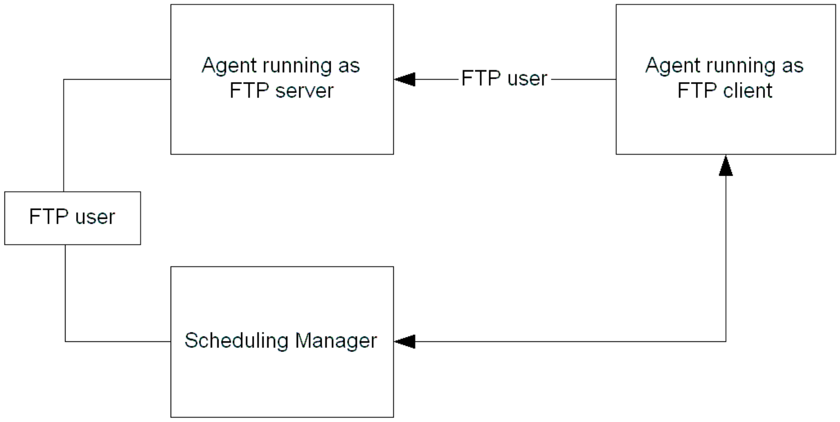
- Download an ASCII or binary file from a remote FTP server to your agent computer
- Upload an ASCII or binary file from your agent computer to a remote FTP server

You can set up the agent to run as an FTP client, FTP server, or both.

The following diagram shows you the relationships between the agent running as an FTP client, a scheduling manager, and an FTP server.



The following diagram shows you the relationships between the agent running as an FTP server, a scheduling manager, and another agent running as an FTP client.



How to Set Up the Agent as an FTP Client

When you set up the agent as an FTP client, it can log in to remote FTP servers, download files from those servers, and upload files to those servers.

Note: When the agent runs as an FTP client only, other FTP clients (such as other agents) cannot log in to the agent to FTP files. To allow other FTP clients to log in and transfer files, set up the agent to run as an FTP server.

To set up the agent as an FTP client, follow these steps:

1. Configure the agent as an FTP client (see page 123).
2. Define FTP rules for local security on the agent (see page 125).

   This step only applies if local security is enabled on the agent.
3. Define the FTP User on the scheduling manager (see page 126).

Note: You can also set up the agent to use Secure (SSL) FTP.

Configure the Agent as an FTP Client

You can configure the agent as an FTP client after installation using the following procedure.

To configure the agent as an FTP client

1. Change to the agent installation directory.
2. Stop the agent.
3. Open the agentparm.txt file.
4. Add the plugins.start_internal_*n* parameter for the FTP plug-in. The *n* suffix must increase sequentially for each agent plug-in.

   Example: plugins.start_internal_5=ftp

5. Define the following parameter if you are using Internet Protocol version 6 (IPV6):

   ```
   ftp.passive=true
   ```

6. (Optional) Define the following parameters:

   ftp.ascii.ccsid

   Defines the Coded Character Set Identifier (CCSID) to use for ASCII file transfers. If the file being transferred exists on the target computer, the file is written using the encoding of the existing file.

   Default: 819

   ftp.client.updatemsg

   Defines the frequency interval in milliseconds in which the status information for an FTP job in EXEC state is updated.

   Default: 30 000 (30 seconds)

   ftp.data.compression

   Specifies whether to compress data for all FTP jobs on this agent for transfer. The value ranges from zero (0) for no compression to nine (9) for the best compression. If the compression level is also specified in the job definition, the ftp.data.compression value is ignored, and the data is compressed using the level specified in the job definition. To use FTP data compression, the agent must run both FTP client and FTP server.

   Default: 0 (no compression)

ftp.download.owner

Specifies a default user ID on the computer where the agent is installed. This user ID determines the access permissions of a downloaded file on the agent computer. When the file is downloaded, the file is created with this user as the file owner.

Notes:

- This parameter only applies to agents installed on UNIX systems.
- A password for the local user ID is not required on the scheduling manager.

ftp.passive

Specifies whether the agent FTP client uses a passive mode connection, as follows:

- false—The agent uses an active mode connection.
- true—The agent uses a passive mode connection.

Default: false

Note: We recommend you set the value to true under any of the following conditions: the agent uses IPV4 and the FTP server uses IPV6 for communication, the FTP server resides beyond the firewall, and/or the FTP server opens a listening port for the data channel.

7. Save and close the agentparm.txt file.
8. Start the agent.

   The agent is configured as an FTP client.

Configure the Agent FTP Client to Use Secure Copy Protocol (SCP)

You can configure the agent to act as an FTP client that uses the secure copy protocol (SCP) to transfer binary files.

To configure the agent for secure copy file transfers, configure the following agent parameters on the agent:

ftp.client.updatemsg

Specifies the status update interval in milliseconds (ms).

Default: 30000 (ms)

ftp.download.owner

Specifies a default user ID on the computer where the agent is installed. This user ID determines the access permissions of a downloaded file on the agent computer. When the file is downloaded, the file is created with this user as the file owner.

Notes:

- This parameter only applies to agents installed on UNIX systems.
- A password for the local user ID is not required on the scheduling manager.

ftp.scp.sshd.timeout

Controls the timeout, in milliseconds (ms), for SCPv2.

Default: 30000 (ms)

ftp.scp.debug.enable

Sets whether debugging of the secure copy protocol (SCP) sessions is enabled. The output is stored in the ftp_scp_debug.log.

- false—Disables debugging.
- true—Enables debugging.

Default: false

Define FTP Rules for Local Security on the Agent

By default, the agent security file denies all users from issuing FTP-related commands on the agent computer. To allow users to issue FTP-related commands while local security is enabled, define FTP rules in the security.txt file.

Note: Local security is enabled on the agent if the security.level parameter is set to on in the agentparm.txt file.

Example: Restrict FTP Access to Files in a Specified Directory

In this example, FTP users that log into the FTP server do not have access to the files in the /local/pub directory and its subdirectories.

```
f a * * +
f d * * /local/pub/+
```

Example: Allow FTP Access to Only One Directory

In this example, FTP users that log into the FTP server have access to the /local/pub directory and its subdirectories but do not have access to any other directories.

```
f d * * +
f a * * /local/pub
f a * * /local/pub/*
```

More information:

Security File Rules (see page 114)
Refresh an Agent Security File (see page 119)

Define the FTP User on the Scheduling Manager

To use the agent as an FTP client, define each FTP user on the scheduling manager.

For more information about defining users, see the documentation for your scheduling manager.

How to Set Up the Agent as an FTP Server

The agent supports a built-in FTP server capability. You can enable the agent to act as a generic FTP server in addition to its other roles. This server comes under the security rules established for the agent.

To set up the agent as an FTP server, follow these steps:

1. Configure the agent as an FTP server (see page 127).
2. Set up local security on the agent (see page 112).

   Note: Local security must be enabled (the security.level parameter in the agentparm.txt file must be set to on). If the agent runs as an FTP server, clients can log in to the agent and transfer files.
3. Define the FTP user on the agent (see page 128).

   Note: The FTP user ID used to connect to the agent running as an FTP server must be defined on that agent and the scheduling manager.

The agent as an FTP server can handle ASCII and binary transfers, wildcard requests, simple GET and PUT requests for single files, and MGET and MPUT requests for multiple files. The agent has a secure store of FTP server user IDs and associated passwords. The ftpusers.txt file, located in the directory that contains the agent program files, stores these user IDs and their corresponding hashed passwords.

The agent running as an FTP server does not support anonymous FTP requests. For audit purposes, the agent provides a detailed log of all FTP requests.

Configure the Agent as an FTP Server

You can configure the agent as an FTP server while installing the agent or after installation. However, to set up optional FTP server features, modify the agentparm.txt file after installation.

To set up the agent as an FTP server

1. Change to the agent installation directory.
2. Stop the agent.
3. Open the agentparm.txt file.
4. Add the plugins.start_internal_*n* parameter for the FTP plug-in. The *n* suffix must increase sequentially for each agent plug-in.

   Example: plugins.start_internal_5=ftp

5. Define the following parameters:

   ftp.noserver

   Specifies whether the agent FTP server is enabled or disabled. If ftp.noserver is set to false, the FTP server is enabled. If the ftp.noserver is set to true, the FTP server is disabled.

   Default: true

   ftp.serverport

   Specifies the port number for the agent to act as an FTP server.

   Default: 21

   Limits: 1-65534

6. (Optional) Define the following parameter:

   ftp.ascii.ccsid

   Defines the Coded Character Set Identifier (CCSID) to use for ASCII file transfers. If the file being transferred exists on the target computer, the file is written using the encoding of the existing file.

   Default: 819

7. Save and close the file.
8. Start the agent.

   The agent is configured as an FTP server.

Set Up Local Security on the Agent

To use the agent as an FTP server, set up local security on the agent.

To set up local security on the agent, follow these steps:

1. Enable local security (see page 112).
2. Configure the security.txt file (see page 113).
3. Refresh the security.txt file (see page 119).

Define the FTP User on the Agent

To run FTP workload through an agent operating as an FTP server, define the FTP user ID and the corresponding password on the agent. The FTP user ID belongs to the user authorized to make the file transfer.

To define FTP users on the agent, run the ftpusrcfg utility located in the agent installation directory.

Note: If you set up the agent as an FTP server during installation, you defined one FTP user ID and password. Use the ftpusrcfg utility to define additional FTP users or change the password of an FTP user.

FTP Server Maintenance

The agent FTP server is a fully functional FTP server with user authentication support. To maintain the FTP server, manage the FTP users file and configure local security on the agent.

The ftpuser.txt file, located in the directory that contains the agent program files, stores FTP user IDs and passwords. The ftpusers.txt file uses one line for each entry with the user ID in the first position followed by the hashed password.

Manage FTP User IDs and Passwords

To run FTP workload on an agent operating as an FTP server, define the FTP user ID and password on the agent. You use the ftpusrcfg utility to add, delete, and change FTP user IDs and passwords on agents operating as FTP servers. Changes made with the utility update the ftpusers.txt file. Restart the agent for the changes to take effect.

To manage FTP user IDs and passwords

1. Change to the agent installation directory.
2. Enter the following command from a command line on the same computer as the agent:
   - On UNIX, enter ftpusrcfg -a|-d|-m|-l *userID password*
   - On Windows, enter ftpusrcfg.bat -a|-d|-m|-l *userID password*

   -a

   Adds a new user ID. Use with the *userID* and *password* parameters. Enter the user ID first followed by the password.

   -d

   Deletes the specified user ID. Use with the *userID* parameter.

   -m

   Changes the password for the specified user ID. Use with the *userID* and *password* parameters. Enter the userID first followed by the password.

   -l

   Lists all entries in the ftpuser.txt file. The utility does not show passwords in plain text.

   userID

   Specifies the FTP user ID you want to add, change, or delete.

   password

   Specifies the password corresponding to the FTP user ID. Passwords are case sensitive.

   Note: Issuing the ftpusrcfg command without a parameter displays a list of options.
3. Restart the agent if you changed the ftpusers.txt file.

Example: Add a New FTP User ID on UNIX

The following command adds the FTP user ID P01Prod01 with the password cyber:

```
ftpusrcfg -a P01Prod01 cyber
```

Example: Change the Password on Windows for an Existing FTP User

The following command changes the password for the FTP user ID P01Prod01 from cyber to r6ut09:

```
ftpusrcfg.bat -m P01Prod01 r6ut09
```

Configuring SSL FTP

To run FTP workload using Secure Sockets Layer (SSL) communication, enable and configure SSL on the FTP server and the FTP client. When you select to enable SSL FTP during the agent installation process, the installation program does the following:

- Defines default SSL server and client parameters in the agentparm.txt file.
- Adds cacerts, serverkeystore default certificates, a password, and a customized java.security file to the agent installation directory.

You can use the default certificates and settings to configure SSL FTP. You can also generate a certificate and apply your own settings to configure SSL FTP.

Configure SSL FTP Using the Default Certificates and Settings

The default certificates and settings provided during the agent installation let you set up SSL FTP without generating your own certificates and settings.

To configure SSL FTP using the default certificates and settings

1. In the SSL FTP server directory, export the certificate used by the SSL FTP server, for example:

   ```
   jre\bin\keytool -export -alias agentname -file key.cer -keystore serverkeystore
   ```

   agentname

   Specifies the name of the agent.

   You are prompted for a password. The default password is cyberuser.

   Note: You require the alias “agent” to ensure that you use the certificate provided by the agent.

2. Copy the created file, in this case, key.cer, to the SSL FTP client directory if it is different from the server directory.

3. Import the created file, in this case, key.cer, into the truststore file supplied by Sun (cacerts), for example:

   ```
   D:\agent\R6SP2>jre\bin\keytool -import –alias agentname -file key.cer -keystore
   cacerts
   Enter keystore password: changeit
   Owner: CN=cyberuser, OU=ESP System Agent, O=Cyber, L=Markham, ST=Ont, C=CA
   Issuer: CN=cyberuser, OU=ESP System Agent, O=Cyber, L=Markham, ST=Ont, C=CA
   Serial number: 4152d5dc
   Valid from: Thu Apr 21 09:55:40 EDT 2005 until: Mon Jun 05 09:55:40 EDT 2006
   Certificate fingerprints:
   MD5: 74:F2:17:20:B6:B0:10:AE:AC:88:9A:BA:AA:3A:6D:73
   SHA1:4C:88:B6:39:64:65:98:AD:3E:1E:33:05:12:13:9C:4A:F4:4E:E7
   :FA
   Trust this certificate? [no]: yes
   Certificate was added to keystore
   ```

4. Start the agent acting as an FTP server.

   Note: If another agent acts as an FTP client, copy the created key.cer to the SSL FTP client directory and import the certificate.

5. Run a test upload and download job to verify the setup.

   For more information about scheduling FTP workload, refer to the documentation for your scheduling manager.

How to Configure SSL FTP Using a Generated Certificate

This process configures SSL FTP using user-generated certificates and settings.

To configure SSL FTP using a generated certificate, follow these steps:

1. Generate a server keystore (see page 132).
2. Verify the server keystore (see page 133).
3. Encrypt the server keystore password (see page 133).
4. Configure an SSL-enabled FTP server on the agent (see page 134).
5. Add the certificate to the client keystore on the agent (see page 135).
6. Configure an SSL-enabled FTP client on the agent (see page 137).

Note: After you configure SSL FTP, you can enable and disable it as required. If SSL is disabled on the FTP client after configuration and you want to run SSL FTP workload, you can specify SSL in the job definition instead.

Generate a Server Keystore Using a Generated Certificate

To configure an SSL-enabled FTP server using user-generated certificates and settings, generate a keystore. You can generate your own keystore using the keytool utility provided with the JRE. The utility is located in the JRE/bin directory.

Note: Add the path to the keytool to your path variable.

To generate a server keystore

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

   ```
   Jre/bin/keytool -genkey -alias agent1 -keystore ./serverkeystore1
   ```

4. Follow the prompts.

   Note: Encrypt the keystore password you enter. Use the password utility to encrypt the keystore password.

5. Edit the agentparm.txt file for the following parameters:

   ```
   ftp.server.ssl.keystore=agent_install/serverkeystore1
   ftp.server.ssl.keystore.password=encrypted_password
   ```

6. Save the agentparm.txt file.
7. Start the agent.

Example: Generate a Server Keystore

The following example shows sample keytool prompts and values:

```
/home/ESP_System_Agent_R7>keytool -genkey -alias agent -keystore ./serverkeystore
Enter keystore password:  123456
What is your first and last name?
  [Unknown]:  Cyberuser
What is the name of your organizational unit?
  [Unknown]:  agent
What is the name of your organization?
  [Unknown]:  Cybermation
What is the name of your City or Locality?
  [Unknown]:  Markham
What is the name of your State or Province?
  [Unknown]:  Ontario
What is the two-letter country code for this unit?
  [Unknown]:  CA
Is CN=Cyberuser, ESPSystemAgent, O=Cybermation, L=Markham, ST=Ontario, C=CA correct?
  [no]:  yes
Enter key password for [set AGENT value for your book](RETURN if same as keystore
password):
```

Verify a Server Keystore

Use the keytool utility to verify the accuracy of the server keystore you generated.

To verify a server keystore

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

   ```
   keytool -list -v -keystore serverkeystore
   ```

3. Follow the prompts.

   Information about the server keystore is displayed.

Example: Verify a Server Keystore

The following example shows sample keytool prompts and values:

```
/home/ESP_System_Agent_R7>keytool -list -v -keystore serverkeystore
Enter keystore password:  123456
Keystore type: jks
Keystore provider: SUN
Your keystore contains 1 entry
Alias name: agent
Creation date: Apr 21, 2005
Entry type: keyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Cyberuser , ESPSystemAgent, O=Cybermation, L=Markham, ST=Ontario, C=CA
Issuer: CN=Cyberuser, ESPSystemAgent, O=Cybermation, L=Markham, ST=Ontario, C=CA
Serial number: 4123a631
Valid from: Thu Apr 21 09:55:40 EDT 2005 until: Mon Jun 05 09:55:40 EDT 2006
Certificate fingerprints:
         MD5:  39:D8:D9:4F:50:1C:43:A2:27:4D:50:75:32:E9:9D:40
         SHA1: 98:30:54:C0:F7:4E:34:FF:DC:0A:85:D8:F7:98:D6:B7:41:7D:E7:58
```

Encrypt a Password for the Server Keystore

You require an encrypted keystore password to configure an SSL-enabled FTP server on the agent. Use the Password utility provided with the agent to encrypt the keystore password you used when you generated the keystore.

Change an SSL FTP Server Keystore Password

To change an SSL FTP server keystore password

1. Enter the following command:

   ```
   ./cybAgent -s
   ```

   The agent stops running.

2. Open the agentparm.txt file.
3. Define the following parameter:

   ftp.server.ssl.keystore.password

   Specifies the new encrypted password.

4. Save and close the file.
5. Enter the following command:

   ```
   ./cybAgent
   ```

   The agent starts running. The password is changed.

Configure an SSL-enabled FTP Server on the Agent

You use the generated server keystore and its encrypted password to configure an SSL-enabled FTP server on the agent.

To configure an SSL-enabled FTP Server on the agent

1. Change to the agent installation directory.
2. Stop the agent.

   The agent stops running.

3. Open the agentparm.txt file.
4. Set the following parameters:

   ```
   security.level=on
   ftp.noserver=false
   ftp.server.ssl=true
   ```

5. Specify the following parameters:

   ftp.server.ssl.keystore

   Specifies the full path of the keystore file. The default file name is serverkeystore. You can use keytool, provided with the JRE, to create your own keystore.

   Example: ftp.server.ssl.keystore=/R7/serverkeystore

ftp.server.ssl.keystore.password

Specifies the encrypted password for the server keystore that contains an X509 certificate. This password is sent to the client during the handshake process. The default password is cyberuser (encrypted).

Note: You can use the agent password utility to encrypt your password before using it in the agentparm.txt file.

6. Save and close the agentparm.txt file.
7. Start the agent.

   The agent starts running and the FTP server on the agent is SSL-enabled.

How to Add a Certificate to the Client Keystore on the Agent

You add a certificate to the client keystore to configure an SSL-enabled FTP client on the agent.

To add a new certificate to the client keystore on the agent, follow these steps:

1. Export the certificate from the server keystore (see page 135).
2. Import the certificate to the client keystore on the agent (see page 136).
3. Verify the client keystore on the agent (see page 137).

Export the Certificate from the Server Keystore

You can export the certificate from the server keystore using the keytool utility provided with the JRE.

Note: Add the path to keytool to your path variable.

To export the certificate from the server keystore

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

   ```
   keytool -export -file key.cer -keystore serverkeystore
   ```

   Note: To export the certificate generated with an alias, include the same alias in the export command. For example, suppose a certificate was generated with the following command:

   ```
   keytool -genkey -alias agent -keystore ./serverkeystore
   ```

   To export that certificate, use the following command:

   ```
   keytool -export -alias agent -file key.cer -keystore serverkeystore
   ```

3. Follow the prompts.

   The server keystore certificate is exported.

Example: Export the Certificate from the Server Keystore

The following example shows sample keytool prompts and values:

```
/home/ESP_System_Agent_R7>keytool -export -file key.cer -keystore serverkeystore
Enter keystore password:  654321
Certificate stored in file <key.cer>
```

Import a Certificate to the Client Keystore on the Agent

You can import the server keystore certificate to the client keystore on the agent using the keytool utility provided with the JRE.

Note: Add the path to keytool to your path variable.

To import a certificate to the client keystore on the agent

1. Change to the directory that contains the agent program files.
2. Enter the following command:

   ```
   keytool -import -file key.cer -keystore cacerts
   ```

   To import a certificate that was exported with an alias, include the same alias in the import command. For example, suppose a certificate was exported with the following command:

   ```
   keytool -export -alias agent -file key.cer -keystore serverkeystore
   ```

   To import that certificate, use the following command:

   ```
   keytool -import -alias agent -file key.cer -keystore cacerts
   ```

3. Follow the prompts.

   The certificate is imported to the agent client keystore.

Example: Import a Certificate to the Client Keystore on the Agent

The following example shows sample keytool prompts and values:

```
C:\Program Files\Cybermation\ESP System Agent>keytool -import -file key.cer
-keystore cacerts
Enter keystore password:  changeit
Owner: CN=Cyberuser C, OU=ESPSystemAgent, O=r, L=g, ST=d, C=ca
Issuer: CN=Cyberuser C, OU=ESPSystemAgent, O=r, L=g, ST=d, C=ca
Serial number: 41239e39
Valid from: Thu Apr 21 09:55:40 EDT 2005 until: Mon Jun 05 09:55:40 EDT 2006
Certificate fingerprints:
MD5:  31:CC:29:0F:B6:C8:E9:3C:70:C7:6B:6C:AD:B7:00:38
SHA1:9D:86:A7:51:15:9E:B1:D3:E7:3B:59:C6:B2:E0:E0:3F:3D:C6:97:6
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

Verify a Client Keystore on the Agent

You can verify the keystore on the agent using the keytool utility provided with the JRE.

To verify a client keystore on the agent

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

   ```
   keytool -list -v -keystore cacerts
   ```

3. Follow the prompts.

   Information about the clientkeystore is displayed.

Example: Verify a Client Keystore

The following example shows sample keytool prompts and values:

```
C:\Program Files\Cybermation\ESP System Agent>keytool -list -v -keystore cacerts
Enter keystore password:  changeit
Keystore type: jks
Keystore provider: SUN
Your keystore contains 26 entries
Alias name: equifaxsecureebusinessca1
Creation date: Apr 21, 2005
Entry type: trustedCertEntry
Owner: CN=Equifax Secure eBusiness CA-1, O=Equifax Secure Inc., C=US
Issuer: CN=Equifax Secure eBusiness CA-1, O=Equifax Secure Inc., C=US
Serial number: 4
```

Configure an SSL-enabled FTP Client on the Agent

If you use the agent FTP client to connect to the SSL-enabled FTP server on the agent, configure the FTP client for SSL communication.

To configure an SSL-enabled FTP client on the agent

1. Change to the agent installation directory.
2. Stop the agent.

   The agent stops running.

3. Open the agentparm.txt file.

4. Set the following parameter:

   ```
   ftp.client.ssl=true
   ```

5. Specify the following parameter:

   ftp.client.ssl.truststore

   Specifies the full path name of the truststore file. The default file name is cacerts. You can use keytool, provided with the JRE, to create your own truststore.

6. Save and close the agentparm.txt file.

7. Start the agent.

   The agent starts running and the FTP client on the agent is SSL-enabled.

Chapter 11: Maintaining Spool and Log Files

This section contains the following topics:

Spool File Maintenance

The output for workload is stored in spool files that the agent software generates. Depending on the type of workload the agent runs, the spool files are stored in and accessed from different locations.

Spool files are limited in size by the available space on the file system where they reside. We recommend that you clear spool files regularly to maintain storage space. The agent does not clear the spool files by default. You can configure the agent to clear spool files automatically.

Configure the Agent to Clear Spool Files Automatically

You can configure the agent to clear the UNIX workload spool files automatically by modifying the agentparm.txt file. You can also set parameters to specify a file expiration time and sleep time.

Note: The agent logs the spool-file cleanup activity in the runner_spool_cleaner.log log, located in the log directory of the agent.

To configure the agent to clear spool files automatically

1. Change to the agent installation directory.
2. Stop the agent.
3. Open the agentparm.txt file located in the agent installation directory.

4. Set the following parameter:

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

5. (Optional) Specify the following additional parameters:

   runnerplugin.spool.expire

   Specifies the file expiration time. The agent deletes spool files that are older than this value. This parameter uses the following syntax:

   <*n*><D|H|M|S>

   n

   D

   H

   M

   S

   Default: 10D (10 days)

   Note: You cannot specify combinations of time periods. For example, 12D3H is not

   valid. If you specify a number only, the agent assumes days by default.

   runnerplugin.spool.sleep

   Specifies the sleep interval. At every interval, the agent checks for spool files that

   meet the expiration time and deletes them.

   Default: 1D (1 day)

6. Save and close the agentparm.txt file.

7. Start the agent.

   The agent is configured to clear spool files automatically.

Example: Delete Spool Files Older Than 10 Days

Suppose that you want to configure the agent to review the spool files every 36 hours and delete spool files that are older than 10 days.

Add the indicated values to the following parameters in the agentparm.txt file:

```
runnerplugin.spool.clean.enable=true
runnerplugin.spool.expire=10D
runnerplugin.spool.sleep=36H
```

The agent deletes spool files that are older than 10 days.

Example: Check Spool Files When the Sleep Interval Is Greater Than the File Expiration Time

Suppose that you want to configure the agent to review the spool files every 50 minutes and delete spool files that are older than 50 minutes as specified by runnerplugin.spool.expire.

Add the indicated values to the following parameters in the agentparm.txt file:

```
runnerplugin.spool.clean.enable=true
runnerplugin.spool.expire=50M
runnerplugin.spool.sleep=2H
```

The agent ignores the two hour sleep interval set by runnerplugin.spool.sleep.

More information:

Configure Agent Parameters on the Agent (see page 56)

Configure the Agent to Automatically Delete Spool Files of Completed Jobs

You can configure the agent to delete a spool file automatically when the job completes successfully.

To configure the agent to delete spool files automatically, add the following parameter to the agentparm.txt file and restart the agent:

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

Clear Windows Spool Files Using the Clearspool Command

On Windows, you can clear agent spool files that are older than a specific number of days using the clearspool command. When you issue the command, you can also view debugging messages as the command runs.

To clear Windows spool files using the clearspool command

1. Define the ESPAGENTDIR environment variable with the path to agent installation directory.

   The agent installation directory must contain a valid agentparm.txt file.

2. Enter the following command at the Windows command prompt:

   ```
   clearspool n [debug]
   ```

   n

   > Specifies the maximum number of days a spool file is maintained. The clearspool command removes all files older than n days.

   debug

   > Optional. Displays messages to the command prompt as the clearspool command runs.

Example: Clearing spool files older than five days

The following command deletes all files older than five days.

```
clearspool 5
```

Example: Displaying the debugging messages

The following command deletes all files older than 10 days and displays debugging messages to the command prompt as it runs.

```
clearspool 10 debug
```

Log File Maintenance

The agent keeps a set of logs that you must clear periodically to maintain disk space availability. The log files contain records of all messages between the agent and the scheduling manager, and internal messages. These files are located in the log directory by default and are updated continually while the agent is running. The types and number of logs that are generated depend on the log.level parameter set in the agentparm.txt file.

You can configure agent log file properties that control the log file size, the types and number of log files that are generated, and how the agent archives the log files. Depending on your scheduling manager, you can also clear log files manually.

Configure the Agent to Clear Log Files Automatically

The agent has a housekeeping function that automatically removes all existing files with the extension .log that reach a certain size. You can configure the agent to clear the log files automatically by modifying the agentparm.txt file.

To configure the agent to clear log files automatically

1. Change to the agent installation directory.
2. Stop the agent.
3. Open the agentparm.txt file located in the agent installation directory.
4. Edit the following parameter to specify the maximum log size (in bytes).

   ```
   log.maxsize
   ```

   When the log file exceeds the specified size, the agent archives it and starts a new log file.

5. Edit the following parameter to specify the log archiving options:

   ```
   log.archive
   ```

   Note: The agent ignores the log.maxsize value if the log.archive parameter is set to 3. The agent does not create an archive file, but appends new log entries to the current logs.

6. Edit the following parameter to specify the types of logs and number of logs to generate:

   ```
   log.level
   ```

   Note: Level 2 is adequate for general, initial testing, and level 0 is adequate for production unless problems arise requiring more details for troubleshooting.

7. Save and close the agentparm.txt file.
8. Start the agent.

   The agent is configured to clear log files automatically.

   Note: In some combinations of log.level and log.archive settings, a new file is generated (runner_plugin_transmitter_queue.log).

More information:

Agent Parameters in the agentparm.txt File (see page 57)

Enable or Disable Job Logs

By default, the agent creates a job log for every script or binary request that runs on the system it manages. The job log contains environment and other diagnostic information that you can use to debug failed jobs.

To enable or disable job logs, edit the following parameter in the agentparm.txt file and restart the agent:

oscomponent.joblog

Sets whether the agent creates a job log for each job that runs.

- false—Disables job logs
- true—Enables job logs

Default: true

Note: The agent stores the job logs in the spool file directory, which you must clear periodically depending on the volume of your workload. You can also configure the agent to delete job logs automatically when jobs complete successfully.

More information:

Configure the Agent to Automatically Delete Job Logs (see page 144)

Configure the Agent to Automatically Delete Job Logs

You can configure the agent to delete a job log automatically when the job completes successfully.

To configure the agent to delete job logs automatically, add the following parameter to the agentparm.txt file and restart the agent:

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

Chapter 12: Troubleshooting

This section contains the following topics:

Contacting Product Support Services

The sections in this chapter can help you perform basic troubleshooting procedures.

Review any error resolution with your UNIX or Windows system administrator. If this information does not help you resolve the problem, contact Product Support Services.

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

Display Debugging Information During Agent Installation

If you have problems with the agent installation, you can display debugging information on your system to use for troubleshooting.

To display debugging information during agent installation

- On UNIX, enter the following commands before you run setup.bin:

  ```
  LAX_DEBUG=true
  export LAX_DEBUG
  ```

- On Windows, press the Ctrl key and double-click setup.exe.

Your system opens a separate text screen during the agent installation process that displays debugging information.

Collect Log Files for Agents Running on UNIX

During service request investigations, Product Support Services commonly requires log files to help resolve a problem. The following procedure writes the jar file to the /tmp directory. You can create the jar file anywhere you want.

This procedure makes the following assumptions:

- Your PATH environment variable defines the path to the Java bin directory.
- You know the path of the directory the agent is installed in.
- You have permissions required to traverse the directories specified in the procedure.
- The find command on your system is the standard one.

If you need the required permissions or the path to Java on your machine, see your system administrator.

To collect log files for agents running on UNIX

1. Create a temporary directory for the jar file. For example, if the Service Request number is 12345, type the following command:
   ```
   cat /dev/null > /tmp/sr12345_logfiles.jar
   ```
2. Create the jar file. For example, type the following command on one line:
   ```
   find /<AgentInstallDirectory>/log -type f -mtime -2 -exec jar uvf
   /tmp/sr12345_logfiles.jar {} \;
   ```
   Note: The find command above limits the amount of data included in the jar file using the -mtime switch. In the above example, -mtime -2 includes all log files whose last modified data is within the last two days.
3. FTP the jar file, using binary mode, to a machine where you can email CA.
4. Email the jar file to CA. Check that the jar file name includes the service request name. Identify the service request number on the subject line of your email.

   Note: Emails sent to CA cannot exceed 5 megabytes. If the file is greater than 5 megabytes, please contact the Product Specialist investigating your issue for assistance.

Collect Log Files for Agents Running on Windows

During service request investigations, Product Support Services commonly requires log files to help resolve a problem. The following procedure creates a jar file containing the required log files. The Jar command used to create the jar file comes with the standard distribution of Java.

To collect log files for agents running on Windows

1. Open the Windows command Prompt.
2. Find the path to the agent installation directory:

   ```
   dir /b /s | findstr /s /i cybagent.exe
   ```

3. Find the path to the jar command:

   ```
   dir /b /s | findstr /s /i jar.exe
   ```

4. Create temporary directories. For example, if the service request number is 98765, type the following commands:

   ```
   mkdir C:\sr98765
   cd C:\sr98765
   mkdir logfiles
   ```

5. Xcopy the agent log files to the temporary directory. For example, type the following commands:

   ```
   xcopy "<AgentInstallDirectory>\log\*" C:\sr98765\logfiles /S /q
   ```

6. Create the jar file. For example, type the following commands:

   ```
   jar cf C:\temp\sr98765_logfiles.jar C:\sr98765\logfiles
   ```

7. Email the jar file to CA. Check that the jar file name includes the service request name. Identify the service request number on the subject line of your email.

   Note: Emails sent to CA cannot exceed 5 megabytes. If the file is greater than 5 megabytes, please contact the Product Specialist investigating your issue for assistance.

Using a Job Log to Debug a Failed Job

By default, the agent creates a job log for every script or binary request that runs on the system it manages. The job log contains environment and other diagnostic information that you can use to debug failed jobs.

The agent stores each job log in the spool file directory using the following naming convention:

```
job.hash.joblog
```

job

Specifies the name of the job.

hash

Specifies the encryption key that the agent uses to encrypt messages.

Example: Using a Job Log to Debug a Failed Job

A job, running on a Windows system, fails. The agent records the following message in the job log named x.E61ADD84CD3C0864D155EEADD4EEECA6D509D23E.joblog.

```
20071125 20342154+0500 . MBAGENT X/Y/Z State SUBERROR Failed SetEnd Status(Error
creating stdin file) Cmpc(20004) JobLogId(E61ADD84CD3C0864D155EEADD4EEECA6D509D23E)
User(MBAGENT) Host(workstation)
```

More information:

Enable or Disable Job Logs (see page 144)

Agent Logs

The agent provides logging facilities to assist in testing and debugging. The logging facilities can specify logging targets, message levels, and buffering processing.

The agent supports log levels 0, 1, 2, 3, 4, and 5, where level 0 provides the least information and level 5 provides the most.

- Levels 0, 1, and 2 create logs of any errors including the receiver and transmitter logs.
- Level 3 adds queues.
- Levels 4 and 5 add debugging information.

When you install the agent, the log level is set to 0 by default.

In a standard agent installation, the agent maintains the log files in a directory named log, which resides in the agent installation directory.

Log File Structure

All log files have the following basic structure:

```
Date Time <Time Zone> <Message priority> <Thread Group>.Thread.Class.method[:line
number] - <message>
```

Note: The runner_os_component.log log file has a slightly different structure.

Example: Log file structure

```
07/06/2009 10:22:32.609 EDT-0400 2 TCP/IP Controller
Plugin.Transmitter.CybTransmitter.run[:129] - Creating the processor pool[2]
```

Setting Log Levels for Troubleshooting

The log level determines the type and number of logs the agent generates and the amount of information contained in a log. To change the log level, you set the value of the log.level parameter in the agentparm.txt file. You can set the following log level values for troubleshooting:

- 5—Adds debugging information. Use log level 5 for setup and initial testing, and diagnosis of problems.
- 8—Adds tracing information. Use log level 8 for troubleshooting communication problems.

Note: These levels are not intended for continuous use.

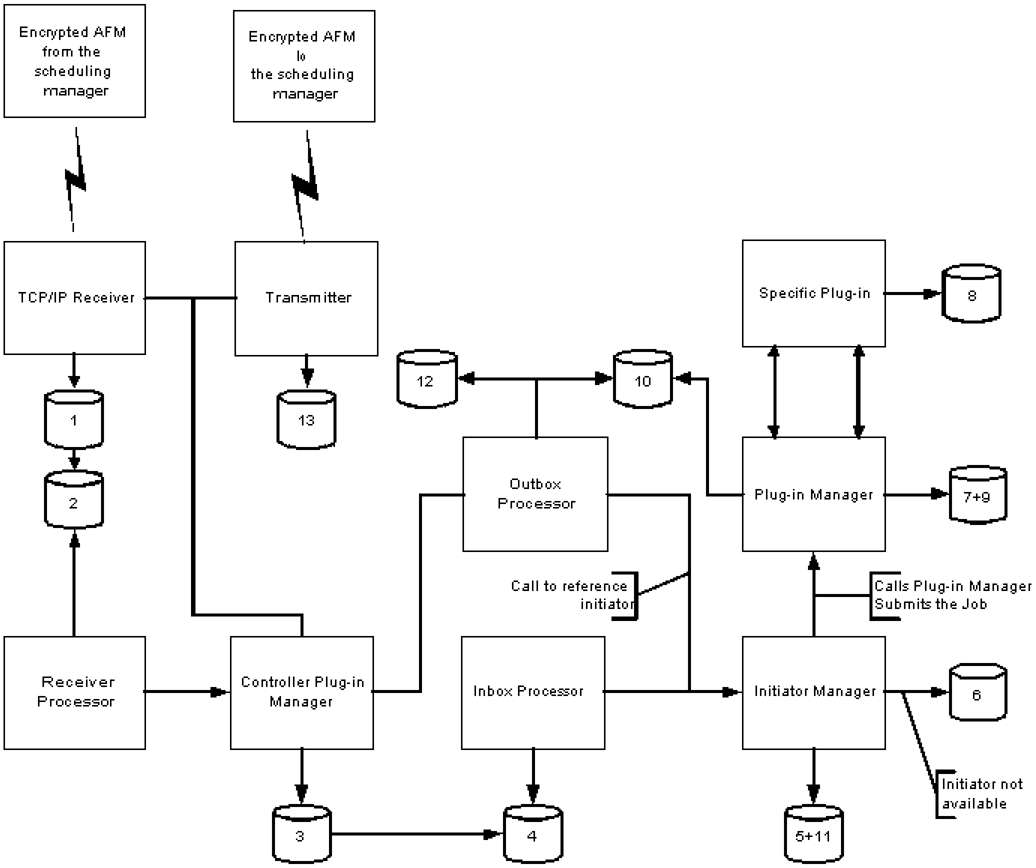
More information:

Agent Parameters in the agentparm.txt File (see page 57)
Configure Agent Parameters on the Agent (see page 56)

Trace an Automated Framework Message (AFM)

The agent logs the path of any AFM as it proceeds from the scheduling manager to the agent. The following dataflow diagram and tables describe how the agent logs the AFMs as it processes them.

The numbers in the data flow diagram refer to the step numbers given in the following table.


Encrypted AFM from the scheduling manager
Encrypted AFM to the scheduling manager
TCP/IP Receiver
Transmitter
Specific Plug-in
8
12
10
1
13
2
Outbox Processor
Plug-in Manager
7+9
Call to reference initiator
Calls Plug-in Manager Submits the Job
Receiver Processor
Controller Plug-in Manager
Inbox Processor
Initiator Manager
6
Initiator not available
3
4
5+11

Step	Log File	Description	Log Level
1	receiver.log	Record of all successfully received AFMs.	0, 1, 2

Step	Log File	Description	Log Level
2	queue_receiver.log	Log for the queue that holds all successfully received AFMs.	0, 1, 2
3	cybrmicontrollerpluginmanager.log	Shows TCP/IP plug-in attempts to send a message to the core.	5
4	queue_inbox.log	All messages from the controller arrive here. Incoming message distributor (inbox) calls initiatormanager to process these messages.	3
5	initiatormanager.log	The initiatormanager records any exception conditions here.	5
6	initiators_waiting_<Job class>.log	If all initiators for this job class are consumed, the job is put in a queue.	3
7	rmipluginmanager.log	Logs the number of active jobs that the plug-in has.	4
8	plug-in specific ■ For runner plug-in, see Runner Plug-in AFM Processing (see page 154). ■ For file monitoring plug-in, see Filemon Plug-in AFM Processing (see page 154).	The message is sent to a plug-in; for example, runner_os_component.log. Once you have completed the trace routine in either of the two other streams, return to the next step in this stream.	-
9	rmipluginmanager.log	Shows the plug-in has attempted to send a message to the core.	4
10	queue_communicator.log	A reply is placed here.	3
11	initiatormanager.log	Shows the initiator has been released.	5
12	messagedistributoroutgoing.log	Shows the message is sent through the controller plug-in manager to the scheduling manager.	5
13	transmitter.log	Log of all sending activity and any errors discovered.	0, 1, 2

Main Stream of AFM Processing

Step	Log File	Description	Log Level
1	receiver.log	Record of all successfully received AFMs.	0, 1, 2
2	queue_receiver.log	Log for the queue that holds all successfully received AFMs.	0, 1, 2
3	cybrmicontrollerpluginmanager .log	Shows TCP/IP plug-in attempts to send a message to the core.	5
4	queue_inbox.log	All messages from the controller arrive here. Incoming message distributor (inbox) calls initiatormanager to process these messages.	3
5	initiatormanager.log	The initiatormanager records any exception conditions here.	5
6	initiators_waiting_<Job class>.log	If all initiators for this job class are consumed, the job is put in a queue.	3
7	rmipluginmanager.log	Logs the number of active jobs that the plug-in has.	4
8	plug-in specific ■ For runner plug-in, see Runner Plug-in AFM Processing (see page 154). ■ For file monitoring plug-in, see Filemon Plug-in AFM Processing (see page 154).	The message is sent to a plug-in; for example, runner_os_component.log. Once you have completed the trace routine in either of the two other streams, return to the next step in this stream.	-
9	rmipluginmanager.log	Shows the plug-in has attempted to send a message to the core.	4
10	queue_communicator.log	A reply is placed here.	3
11	initiatormanager.log	Shows the initiator has been released.	5
12	messagedistributoroutgoing.log	Shows the message is sent through the controller plug-in manager to the scheduling manager.	5
13	transmitter.log	Log of all sending activity and any errors discovered.	0, 1, 2

Runner Plug-in AFM Processing

Step	Log File	Description	Log Level
	internal_plugin_queue_for_runnerplugin.log	All AFMs sent to the Runner plug-in is logged here.	3
	runner_plugin_executing_jobs_map.log	All submitted jobs are logged here.	3
	runner_plugin_transmitter_queue.log	All messages sent to OS component are logged here.	3
	runner_os_component.log	Any errors are logged here.	0, 1, 2
	runner_plugin_receiver_queue.log	All messages coming back from OS component are logged here.	3
	Sent back to core	Return to main stream of AFM processing.	

Filemon AFM Plug-in Processing

Step	Log File	Description	Log Level
1	internal_plugin_queue_for_filemonplugin.log	All AFMs sent to the Filemon plug-in are logged here.	3
2	file_mon_plugin_threads.log	All executing triggers are registered here. When a trigger is completed successfully or has failed, it is removed from this database.	3
3	Sent back to core	Return to main stream of AFM processing.	

Log Resource Usage Information within the JVM

The agent can periodically collect resource usage information within the Java Virtual Machine (JVM) such as memory usage and threads information. This data is logged in a file named simple_health_monitor.log. To log resource usage information within the JVM, edit the following parameters in the agentparm.txt file and restart the agent:

- Set the core.health.monitor.enable parameter to true
- Set the log.level parameter to 5 or greater

You can also specify the polling interval for logging the information using the core.health.monitor.interval parameter. The default is 60 000 ms (1 min). The minimum interval time is 1000 ms (1 sec). If the specified interval is less than the minimum, the agent ignores that value and logs the information at every 1000 ms.

Agent Error Messages on UNIX

This section provides common error messages returned by the agent installed on a UNIX system.

cannot restore segment prot after reloc: Permission denied

Reason:

The Linux computer is SELinux-enabled and prevents the agent from starting.

For more information, see http://www.ittvis.com/services/techtip.asp?ttid=3092.

Action:

Enter the following commands under root authority:

```
find /<AgentInstallDirectory> -name “*.so” –exec chcon -t texrel_shlib_t {} \;
find /<AgentInstallDirectory> -name “*.bin” –exec chcon -t texrel_shlib_t {} \;
chcon -t texrel_shlib_t /<AgentInstallDirectory>/chkusr
```

Note: If the Micro Focus agent plug-in is installed, enter the following command:

```
chcon -t texrel_shlib_t /<AgentInstallDirectory>/cybMFCommand
```

./cybAgent.bin: error while loading shared libraries: libstdc++.so.5: cannot open shared object file: No such file or directory

Reason:

The libstdc++.so.5 library is missing on the Linux computer.

Action:

Install the libstdc++.so.5 file in the /usr/lib directory using the operating system setup routine. For more information, speak to your system administrator.

Note: Do not manually copy the system libraries.

Command does not use shell

Reason:

Only a script can use a shell.

Action:

Remove the shell statement from the job definition.

Command or script name missing

Reason:

In the job definition, you have not defined the required command or script name.

Action:

Add the command name or script name to the job definition.

Command requires a User ID

Reason:

Each command must specify an allowable user.

Action:

Add the USER statement.

Connection aborted by peer: JVM_recv in socket input stream read

Reason:

Reset the communication.timeout parameter to allow more time between a message being sent and an acknowledgement (ACK) being received. When this time is exceeded, the connection is aborted.

Action:

As a starting point, set the value in msecs to 120000 and test. Change the value as needed so this error does not occur.

Error changing directory

Reason:

In the agentparm.txt file, the parameter oscomponent.initialworkingdirectory specifies the working directory. If the path is incorrect, this error occurs.

Action:

1. Verify that the directory exists.
2. Specify the correct path in the parameter.

Error closing stdout, stdout, or stderr

Reason:

An intermittent system problem can cause this error.

Action:

Try to resubmit the job. If this action does not resolve the problem, contact your system administrator.

Error creating pthread

Reason:

System resources are low.

Action:

Contact your UNIX system administrator.

Error creating spool file. Job: JOB.TXT/APPL. xxxx/MAIN, errno: 31, Reason: Too many links

Reason:

On AIX and Linux, the file system can limit the number of spool directories.

Action:

See your operating system documentation for details regarding the spool directory limitations. Clear the spool directories periodically using the agent.

Error creating stdout spool file

Reason:

The user may not have the necessary permissions to create the spool file.

Action:

Verify the user permissions.

Error getting owner of the script

Reason:

Failure to get the owner of the script from the system password file.

Action:

Contact your UNIX system administrator.

Error occurring during submission

Reason:

Connection error with the scheduling manager.

Action:

Verify the connection to the scheduling manager.

Error opening stdin, stdout, or stderr

Reason:

The agent does not have permission to open the file. A system problem can cause this error.

Action:

1. Verify permissions. Change them to allow the agent to open the file.
2. Contact your system administrator.

Error redirecting stdin, stdout, or stderr

Reason:

The agent tries to redirect an input file to another file and an error results. This file could be a spool file or some specified stdin or stdout file. For a stderr message, an intermittent system problem can cause this problem.

Action:

Try to resubmit the job. If this action does not resolve the problem, contact your system administrator.

File not found

Reason:

The agent cannot find the file. Either the wrong path was specified in the job definition, or the file does not exist.

Action:

Verify the path, and ensure that the file exists.

Invalid command: not listed in oscomponent.validcommand

Reason:

Invalid command.

Action:

Add the command to the oscomponent.validcommand parameter in the agentparm.txt file, or contact your UNIX system administrator.

Invalid file name

Reason:

The file path is too long. This error can happen if there are too many symbolic links. The system call returns an error when the name is too long.

Action:

Relocate the files to get rid of the symbolic link.

Invalid shell error

Reason:

In the agentparm.txt file, specify the valid shell in the parameter oscomponent.validshell. The corresponding job definition does not have the correct shell specified in the shell statement.

Action:

1. Verify that the correct shell is specified in the agentparm.txt file. Alternatively, you can set the oscomponent.checkvalidshell=false parameter in the agentparm.txt file, so that the agent does not validate whether the shell is valid.
2. Verify that the corresponding job definition has the correct shell specified in the shell statement.
3. In the first line of a script file, the shell and its path must match exactly what you specified in the oscomponent.validshell parameter.

Irregular file

Reason:

The file is not a regular file, such as an ASCII or binary file. The file may be a directory file or a pipe.

Action:

Determine the file type using ls -l. Replace the file with a regular file.

Not a script file

Reason:

The script contains non-printable characters.

Action:

Use Command instead of Script name in the job definition.

Refused by Agent security

Reason:

The job is refused by the local security on the agent.

Action:

Check the security.txt file.

Script/Command not accessible

Reason:

The symbolic link does not exist.

Action:

Check the definition of the symbolic link to see if it exists.

Script/Command not executable

Reason:

The user does not have the necessary permissions to execute the script or command.

Action:

Verify the user permissions.

Script/Command not readable

Reason:

User does not have read permission.

Action:

Verify the user permissions.

SUID or SGID is missing

Reason:

If the oscomponent.checksuid parameter is set to true in the agentparm.txt file, set the SUID bit to **on** in the script or command.

Action:

Set the SUID bit to **on** or contact your system administrator.

su not found

Reason:

The agentparm.txt file defines the default path. If you place the su command in a different directory than the default, change the oscomponent.subdirectory parameter.

Action:

Change the oscomponent.subdirectory parameter to specify the directory where the su command resides.

User does not exist in the system

Reason:

You did not define the user on the UNIX side.

Action:

Ensure that the user exists on the UNIX side.

Agent Error Messages on Windows

This section provides common error messages returned by the agent installed on a Windows system.

Error 1067 - process terminated unexpectedly

Reason:

1. You do not have sufficient access privileges to run the agent.
2. Agent ports are in use by another process.

Action:

1. You must have a user ID that has administrator authority to run the agent.
2. You have multiple agents trying to use the same ports. The following parameters must be unique for each agent:
   - agentname
   - communication.inputport

In addition to the parameters above, the following parameters must be unique for each agent on Windows:

- oscomponent.servicename
- oscomponent.servicedisplayname

Another process has a port in use that the agent needs. Issue the netstat command to determine what ports are in use. For example,

netstat -na

Another process may be intermittent and may not be accessing the agent ports at the time you issue the netstat command.

Communication Problems Between the Agent and the Scheduling Manager

If there is a communication problem between the agent and the scheduling manger, the jobs are shown in the AGENTDOWN state. The following are possible causes:

- The agent is not started.
- The scheduling manager and the agent have different encryption keys.
- The scheduling manager and the agent have different values for the parameters that must match.
- A firewall is blocking the transmission of the agent responses.
- The scheduling manager address is specified as a DNS name in the agentparm.txt file, but the DNS name resolution on the agent computer is faulty or not available.

Change the Ping Frequency for Contacting an Unresponsive Scheduling Manager

When an agent pings a scheduling manager that is unavailable, the agent goes into a two minute sleep interval by default. Any communication the agent receives during this interval is queued on the agent. After communication is re-established and the sleep interval elapses, the agent sends any queued messages immediately and processes any waiting messages from the scheduling manager.

You can change the sleep interval the agent uses. Lowering this value lets the agent ping a scheduling manager more frequently and re-establish communication quicker.

To change the ping frequency the agent uses to contact an unavailable scheduling manager, add the following parameter to the agentparm.txt file, and restart the agent:

communication.transmitter.senderrordelay

Specifies the sleep interval, in milliseconds, for the agent. Decreasing the time causes the agent to ping an unresponsive scheduling manager more frequently.

Default: 120000 (2 minutes)

Lock the Scheduling Managers Defined on the Agent

A scheduling manager can send a message to the agent to add its connection properties dynamically to the agentparm.txt file. You can configure the agent to lock the scheduling managers defined in its agentparm.txt file and prevent any new scheduling manager additions.

To lock the scheduling managers defined on the agent

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

   ```
   communication.manager.dynamic.modifications.lock=true
   ```

5. Save and close the agentparm.txt file.
6. Start the agent.

   The agent will not allow the addition of any new scheduling managers. Existing scheduling managers defined on the agent are not affected.

Agent Runs Out of Heap Space on Windows

Valid on CA Workload Automation AE

When insufficient heap space is available, the agent shuts down. On Windows, the agent has a default maximum heap size of 64 MB.

The scheduling manager stores spool files and job logs, which accumulate over time resulting in reduced heap space for the agent.

To prevent the heap space from running out, configure the following parameters in the agentparm.txt file to clean up spool and job logs automatically for successfully completed jobs:

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

Problems Starting the Agent on AIX

To run the agent on AIX you must install the Java Runtime Environment (JRE), and set the PATH environment variable.

Note: For the JRE version required for your system, see the *CA Workload Automation Agent for UNIX, Linux, or Windows Release Notes*.

Agent Will Not Start—cybAgent Script is Missing

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

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

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

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

Agent Hangs with Jobs Stuck in Running State

The situation where the agent hangs and jobs are stuck in a running state can occur when the disk space is full where the agent resides. The agent requires some free space to run for its logs, internal databases, spool files, and other artifacts.

To correct this problem, you must free up some space on the file system.

More information:

Configure the Agent to Monitor Available Disk Space (see page 78)

SNMP-related Problems

This section provides error messages related to the SNMP management connector.

CybSnmpPluginDriver terminated: java.net.BindException: Permission denied

Reason:

The SNMP communication port specified by the snmp.trap.listener.port agent parameter is already in use. The agent uses ports below 1024. On UNIX, start the agent as root.

Action:

Ensure that the value specified for the SNMP communication port (snmp.trap.listener.port) is not in use by another application. The default port number is 162.

java.lang.IllegalArgumentException: Passed ip address is invalid

Reason:

SNMP v1 does not support IPv6.

Action:

To ensure that the agent uses IPv4, add the following parameters to the agentparm.txt file:

```
java.net.preferIPv4Stack=true
java.net.preferIPv6Stack=false
```

Windows Job Fails with SUBERROR State and Status 'The environment is incorrect'

Valid on Windows

Jobs that fail with a SUBERROR state and a status 'The environment is incorrect' can be due to the environment variables file size exceeding its limitation of 16 KB.

CPU Monitoring Jobs in a Solaris Zone

For CPU Monitoring jobs to return correct results, verify that your Solaris zone is properly configured. To monitor the CPU usage in a Solaris zone you assign the zone to a resource pool so that the operating system correctly computes load averages.

Note: For information about configuring Solaris zones, search for the *Consolidating Servers and Applications* document on the Solaris Learning website at http://learningsolaris.com.

Problems with Windows Interactive Jobs

The CONTROL CANCEL does not cancel the Windows job that is interactive and uses terminal session other than console.

The CHASE command can report a PID as non-running due to the process not being part of the process tree of which the agent is a root.

When using Windows Interactive jobs, the job stdout and stderr is not redirected to the spool file due to Windows security restrictions.

FTP Job Failure Messages

If an FTP job fails, review the job status and spool file for information. The following status messages can appear when a job fails:

Access is denied

Reason:

The FTP user ID does not have the proper permission to access the file.

Action:

Determine whether you have access to the local path specified in the job definition on the agent computer.

File not found

Reason:

The agent cannot find the file. Either the wrong path was specified in the job definition, or the file does not exist.

Action:

Check that the remote path and file name specified in the job definition exist on the remote server.

Logon unsuccessful

Reason:

A problem with the FTP user exists.

Action:

1. Check that the password of your FTP user ID is correct on the scheduling manager.
2. If the agent runs as an FTP server, check that the user ID and password are also defined on the agent.

Note: If you update the ftpusers.txt file, restart the agent for your changes to take effect.

Password is missing

Reason:

A problem with the FTP user ID exists.

Action:

1. Check that the FTP user ID specified in the job definition is correct.
2. Check that the same FTP user ID is defined on the scheduling manager.
3. If the agent runs as an FTP server, check that the user ID and password are also defined on the agent.

Note: If you update the ftpusers.txt file, restart the agent for your changes to take effect.

Please log in with USER and PASS

Reason:

The FTP server requires SSL enablement.

Action:

Check that the FTP server has SSL enabled.

The system cannot find the path specified

Reason:

The path specified in the job definition is incorrect.

Action:

Check that the local path specified in the job definition exists on the agent computer.

Unknown host

Reason:

The remote server name specified in the job definition is incorrect.

Action:

Check that the remote server name specified in the job definition is correct.

Agent Parameters used for Troubleshooting

You can add the following parameters to the agentparm.txt file, as required, to configure the agent. These parameters are used for troubleshooting.

communication.timeout

Specifies the time, in milliseconds (ms), for TCP/IP that can elapse between a message being sent and an acknowledgement (ACK) being received. If this time is exceeded, the connection is aborted.

Default: 10000 (10 seconds)

communication.transmitter.senderrordelay

Specifies the sleep interval, in milliseconds, for the agent. Decreasing the time causes the agent to ping an unresponsive scheduling manager more frequently.

Default: 120000 (2 minutes)

core.health.monitor.interval

Specifies the polling interval, in milliseconds (ms), for logging the resource usage information within the Java Virtual Machine. You can set this parameter when the core.health.monitor.enable parameter is set to true.

Default: 60000 (1 min)

Note: The minimum interval time is 1000 ms (1 sec). If the specified interval is less than the minimum, the agent ignores that value and logs the information at every 1000 ms.

filemonplugin.sleepperiod

Specifies the time, in milliseconds (ms), a Monitoring job uses as the polling interval for file monitoring. Specify no less than 1000 ms.

Default: 30000 (30 seconds)

Note: This parameter does not apply to CA Workload Automation AE.

ftp.client.separator

Specifies the character that is used to separate multiple file entries in the LOCALFILENAME or REMOTEFILENAME statements.

ftp.ssl.provider=IbmX509

Specifies an AIX parameter that supports IBM JSSE (Java Secure Socket Extension), a Java implementation of SSL and TLS.

Note: Do not change the value.

ftp.userfile=path

Specifies the location of the FTP user ID and password file. The default file name is ftpusers.txt.

Example:

- UNIX: /export/home/userid/WA Agent R11.3/ftpusers.txt
- Windows: C:\\Program Files\\CA\\WA Agent R11.3\\ftpusers.txt

log.allow.method

Specifies whether additional debugging messages, such as class name and line number, are logged.

- false—Does not log additional debugging messages
- true—Logs additional debugging messages in all log files

Default: true

log.archive

Defines the log archiving options:

- 0—Appends current date and time to the log file.
- 1—Renames to logfile.archive and starts a new file.
- 2—Removes current file.
- 3—Appends new log entries to the current logs.

Default: 0

log.folder

Specifies the location of the log files. You can specify the full path to the log file directory or the folder name that stores the log files. If you specify the folder name only, the agent creates the folder in the agent installation directory.

Default: the log subdirectory contained in the agent installation directory

objmon.cpu.scalefactor

Specifies a scale factor to multiply the load averages of a CPU and allow the agent, when processing a CPU Monitoring job, to express the load average as a percentage. This scale factor is for busy computers that would otherwise always report 100 percent use.

Default: 100

Example: If you set the scale factor to 10, and the reported load average is 7, then the reported CPU usage would be 70 percent.

objmon.scaninterval

Specifies the interval, in milliseconds (ms), between successive scans for any Monitoring job that uses continuous monitoring.

Default: 10000 (10 seconds)

Note: A shorter interval puts a greater demand on system resources.

oscomponent.checksuid

Sets whether the agent checks the script setuid or setguid attributes. If set to true, allows execution of the script only if setuid or setguid bit is set.

- false—Does not check the script setuid or setguid attributes
- true—Checks the script setuid or setguid attributes

Default: false

Note: For more information about possible SUID error messages, see Agent Error Messages on UNIX.

oscomponent.dumpenvironment

Specifies whether all environment variables are written to the agent spool file for every RUN job.

- false—Does not write environment variables to the spool file
- true—Writes all environment variables to the spool file

Default: false

oscomponent.libjvmpath

Specifies the path statement to the Java library location.

oscomponent.initialworkingdirectory

Specifies the default initial working directory for all scripts.

- SCRIPT—Sets the path to where the script resides
- USER—Sets the path to the home directory of the owner of the script
- PATH—Sets the path to an absolute path to where the script runs

If you do not specify a value, the parameter defaults to the path where the running cybAgent resides.

Note: You can override the InitialWorkingDirectory on a per-job basis by specifying a value for the PWD environment variable.

oscomponent.noexitcode

Specifies the exit code that tells the agent not to send a completion code to the scheduling manager host.

Limits: 1-255

Default: 255 for UNIX, 127 for Windows

oscomponent.noforceprofile

Specifies whether the agent allows loading a .profile/.login file based on the usual UNIX rules for sourcing .profile/.login. When set to false, if the loginshell is set to false and USER has not been specified in the job definition, no .profile/.login will be sourced.

- false—Allows the agent to load the .profile
- true—Prevents the agent from loading the .profile/.login

Default: false

Note: If oscomponent.loginshell is set to false and USER is not specified in the job definition, no .profile/.login is sourced and oscomponent.noforceprofile is ignored.

oscomponent.noguardianprocess

Specifies whether the agent resumes tracking jobs that were active at the time when the agent is recycled.

- false—Returns the status of any active or inactive job when the agent restarts
- true—Does not return the status of jobs that ran at the time the agent went down. Fails UNIX jobs upon restart and returns the message "Lost Control".

Default: false

Note: To enable the default, set the persistence.coldstart parameter to false or comment it out.

oscomponent.noswitchsuid

Specifies whether the agent verifies the presence of setuid or setguid bits on the script.

- false—The agent will always setuid to the owner of the script.
- true—The agent will setuid to the owner of the script only if the script has the setuid or setguid bit on.

Default: false

Note: If oscomponent.noswitchsuid=true and setuid/setgid bit is not on, then a script runs using the authority of the agent (generally root). In this case, as there is only one agentparm.txt per agent installation, the true value would work for scripts with s (set user or group id) set on. However, with scripts which do not have s set on, the script runs under the authority of the agent.

oscomponent.security.turbo

Specifies whether the agent loads the security.txt file into a fast-loading, binary format when the cybAgent process starts up. This setting applies to run jobs only. FTP jobs have separate security rules. When the agent checks security rules for authorizations, it does so much more quickly when this parameter is set to true. Enabling this feature is useful when you have a large, multi-line security file where the required processing slows system operation.

- false—The agent refreshes security rules each run job.
- true—The agent loads security rules into a binary-formatted file and does not check security rules again unless a manual refresh is issued to the agent.

Default: false

oscomponent.umask

Provides support for the umask command. The three-digit octal code specifies the file and directory permissions that are turned off. The default value, 022, sets the following permissions:

File rw-r--r--

Directories rwxr-xr-x

Note: This parameter only applies to files created by the agent such as a spool or log file.

persistence.coldstart

Specifies whether the agent performs a warm or cold start, as follows:

- false—Performs a warm start. The agent tries to use the existing databases. However, if there is sufficient damage, the agent does not start.
- true—Performs a cold start. All databases are automatically destroyed and new ones opened. No manual intervention is required. This setting is recommended if there is extensive damage to the databases. The agent discontinues job monitoring after a cold start.

  Note: On UNIX systems, the agent continues monitoring after a cold start.

Default: false (agent performs a warm start)

persistence.gcinterval

Specifies the persistent garbage collector interval. The garbage collector is invoked at the end of each transaction and runs at least every N milliseconds.

Default: 10000 (10 sec)

runnerplugin.spool.clean.enable

Specifies whether the agent deletes spool files.

- false—Disables the spool file cleaner
- true—Enables the spool file cleaner

Default: false

Note: If enabled, the agent deletes spool files older than 10 days, or 7 days for CA Workload Automation AE, and checks the spool files every day by default. To specify a different file expiration value, set the runnerplugin.spool.expire parameter. To specify a different sleep interval value, set the runnerplugin.spool.sleep parameter.

Chapter 13: Related Documentation

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

Note: To view PDF files, you must 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

To work with the agent and CA Workload Automation AE r11.3, see the following documentation:

Task	Documentation
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 r11.3, see the following documentation:

Task	Documentation
Configure the scheduling manager to work with the agent	*Desktop Client Admin Perspective Help*
Define jobs	*Desktop Client Define Perspective Help*
Monitor and control jobs	*Desktop Client Monitor Perspective Help*

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

CA Workload Automation EE Documentation

To work with the agent and CA Workload Automation EE r11.3, see the following documentation:

Task	Documentation
Configure the agent to work with the scheduling manager	*CA Workload Automation EE Installation and Configuration Guide*
Define jobs	*CA Workload Automation Agent for UNIX, Linux, or Windows User Guide*
Monitor and control jobs	*CA Workload Automation Agent for UNIX, Linux, or Windows User Guide* *CA Workload Automation EE Operator's Guide*

CA Workload Automation SE Documentation

To work with the agent and CA Workload Automation SE r11.3, see the following documentation:

Task	Documentation
Configure the scheduling manager to work with the agent	*CA Integrated Agent Services Implementation Guide* *CA Workload Automation SE Interface Reference Guide* *CA Workload Automation SE Systems Programming Guide*
Define, monitor, and control jobs	*CA Integrated Agent Services User Guide* *CA Workload Automation SE Interface Reference Guide* *CA Workload Automation SE Database Maintenance Guide* *CA Workload Automation SE Command Reference Guide*

Index