Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address

BMC SOFTWARE INC
2103 CITYWEST BLVD
HOUSTON TX
77042-2827
USA

Telephone  713 918 8800
 800 841 2031

Fax 713 918 8000

Outside United States and Canada

Telephone (01) 713 918 8800
Fax (01) 713 918 8000

© Copyright 1999-2017 BMC Software, Inc.

Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of BMC Software, Inc.

BMC, BMC Software, the BMC logo, the BMC Software logo, and other BMC marks, and the tagline “Bring IT to Life” are the exclusive properties of BMC Software, Inc., or its affiliates or subsidiaries and are registered or may be registered with the U.S. Patent and Trademark Office and in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

For BMC Control-M Products that are licensed on the “per CPU - Full Capacity” unit of measurement and installed in an Amazon Web Services (“AWS”) or Microsoft Azure (“Azure”) cloud environment, a license is required for the total number of CPUs in each AWS or Azure instance upon which the Product is installed or which the Product manages, either remotely or locally. For AWS, one CPU is equivalent to one vCPU, as defined by AWS. For Azure, one CPU is equivalent to up to four Virtual Cores (as defined by Azure), rounded up to the closest multiple of four.

Server Endpoint Licensing

All machines upon which any Control-M component is installed or upon which Control-M managed workload runs must be licensed. This includes Control-M Agent/Agentless platforms onto which one or more application plug-ins are installed but also includes Control-M Agent/Agentless platforms where no application plug-ins are installed. Control-M Agent/Agentless platforms on which jobs are ordered are counted regardless of whether those jobs execute or not. All Server Endpoints are counted, including development, staging, QA, pre-production, test, and production environments.
The licensing guidelines for application plug-ins are as follows:

- **ERP and BI/DI**: The application server(s) upon which Control-M managed processes are executed should be licensed in addition to the Control-M Agent machine(s) (in some cases, this may be the same machine).
- **Databases**: Each database server upon which Control-M managed database related processes are being executed should be counted in addition to the Control-M Agent machine(s).
- **AFT and MFT**: Only the machine(s) upon which the AFT or MFT plug-in is installed should be licensed.
- **Web Services, Java & Messaging**: only the Control-M Agent machine(s) upon which the plug-in is installed should be licensed.
- **Backup**: The Control-M Agent machine(s) where the Backup plug-in is installed and also the hosts which are running the backup server software should be licensed (note that this excludes the client machines for which the Backup Server software is managing actual backup processes except where a backup takes place of the backup server machine itself).
- **Cloud**: only the Control-M Agent machine(s) upon which the plug-in is installed should be licensed.
- **Hadoop**: All machines in each managed Hadoop Cluster should be licensed.

IBM® Tivoli® Business Service Manager, IBM Tivoli Workload Scheduler, IBM Cognos, IBM InfoSphere DataStage, IBM iSeries, IBM Websphere, and AIX® are the trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

UNIX® is the registered trademark of The Open Group in the US and other countries.

Linux is the registered trademark of Linus Torvalds.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

SAP® R/2 and SAP R/3, SAP Business Objects, and SAP NetWeaver are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

**Restricted rights legend**

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Field 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.

**Customer support**

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”

**Support website**

You can obtain technical support from BMC 24 hours a day, 7 days a week at (http://www.bmc.com/support). From this website, you can:
Read overviews about support services and programs that BMC offers

Find the most current information about BMC products

Search a database for issues similar to yours and possible solutions

Order or download product documentation

Download products and maintenance

Report an issue or ask a question

Subscribe to receive proactive e-mail alerts when new product notices are released

Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:<yourSupportContractID>, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

Have the following information available so that Customer Support can begin working on your issue immediately:

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)

- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - System hardware configuration
  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level

- Sequence of events leading to the issue

- Commands and options that you used

- Messages received (and the time and date that you received them)
  - Product error messages
  - Messages from the operating system, such as file system full
  - Messages from related software
License key and password information
If you have questions about your license key or password, contact BMC as follows:

- (USA or Canada) Contact the Order Services Password Team at 800 841 2031, or send an e-mail message to ContractsPasswordAdministration@bmc.com.
- (Europe, the Middle East, and Africa) Fax your questions to EMEA Contracts Administration at +31 20 354 8702, or send an e-mail message to password@bmc.com.
- (Asia-Pacific) Contact your BMC sales representative or your local BMC office.

Third party Software
For the provisions described in the BMC License Agreement and Order related to third party products or technologies included in the BMC Product, see https://docs.bmc.com/docs/display/workloadautomation/Control-M+Workload+Automation+Documentation and click Third-party software (TPS).
Contents

Introduction to SSL ................................................................. 8
Preparations and suggested workflow ........................................ 9

SSL configuration ................................................................. 10
Configuring secure communication between Control-M client applications and the Web Server .......... 11
Reverting back to the HTTP connector ..................................... 12
Enabling SSL between Control-M/EM and Control-M/Server ......... 12
Enabling SSL between a distributed Control-M/EM and Control-M/Server .................................................... 13
Enabling SSL on Control-M/Server .............................................. 14
Enabling SSL on Control-M/Agents ............................................. 15
Enabling SSL between Control-M/EM Server and Control-M add-ons ................................................................. 15
Enabling SSL between a distributed Control-M/EM Server and Control-M add-ons .................................................... 17
Disabling SSL between Control-M/EM Server and Control-M client (Changing back to TCP) ............... 18
Disabling SSL between a distributed Control-M/EM Server and Control-M client (Changing back to TCP) .... 19
Configuring secure communication between the web server and GUI Server .................................................... 19
SSL parameters for JacORB .................................................... 21
Configuring NamingViewer (browser for Naming Service) ........... 21
Configuring communication with LDAP or Active Directory servers using SSL ............................................ 22
Deploying SSL on BMC Components ........................................ 23

Certificates ................................................................. 25
Generating new certificates ...................................................... 25
Generating your own certificate ................................................ 26
Bringing your own certificate ...................................................... 27
Examples of script usage when importing certificates and keys .................................................... 28
Certificate copies for Control-M for z/OS ..................................... 30

Advanced Settings and Configurations ..................................... 31
SSL communication parameters ............................................. 31
Configuring a Control-M/Agent to use SSL ................................. 34
Changing a server-agent connection mode for an existing agent .................................................... 34
Configuring Control-M/EM communication with Control-M/Server to use SSL (managed Control-M instances) .................................................................................................................... 35
Configuring Control-M/EM communication with Control-M/Server to use SSL (unmanaged Control-M instances) .................................................................................................................... 35
Storing certificates for TAO ...................................................... 36
Common SSL error messages........................................................................................................ 38
Configuring NamingViewer (browser for Naming Service) ............................................................... 40
Configuring Control-M/EM API JacORB........................................................................................... 41
Setting up a signed certificate for a non-Java Key Database (KDB) .............................................. 42
Setting up a signed certificate for a Java keystore.......................................................................... 43
Configuring security policies.......................................................................................................... 44
Introduction to SSL

Control-M works with the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, ensuring secure communication between the various Control-M components.

SSL for Control-M authenticates and secures communications between:

- Control-M/Server and Control-M/Agent
- Control-M/Server and Control-M/Enterprise Manager (Control-M/EM) servers
- Control-M/EM server and its clients

In addition, and depending on your setup, you can enable security for the following components:

- Control-M Self Service and Control-M Workload Change Manager (WCM) web apps
- Control-M Workload Archiving server
- Control-M NamingViewer
- Batch Impact Manager
- Control-M/EM Web server
Preparations and suggested workflow

The way you set up security in your Control-M environment largely depends on your setup and your security needs. The following diagram depicts a typical workflow:
SSL configuration

You can apply SSL to your whole environment, or to select components only. The following diagram shows SSL communication flows:

For example, if you want SSL enabled between Control-M/EM Server and clients, the Control-M Web Server must be configured with SSL. For agents, you can choose to secure select agents only.

The following sections explain how to configure SSL on the various components.

- Enabling SSL between Control-M/EM and Control-M/Server (on page 12)
- Enabling SSL on Control-M/Agents (on page 15)
- Enabling SSL on Control-M/Server (on page 14)
- Enabling SSL between Control-M/EM Server and Control-M add-ons (on page 15)
- Configuring secure communication between the web server and GUI Server (on page 19)
- Configuring secure communication between Control-M client applications and the Web Server (on page 11)
- Configuring NamingViewer (browser for Naming Service) (on page 21)
- Configuring communication with LDAP or Active Directory servers using SSL (on page 22)
Configuring secure communication between Control-M client applications and the Web Server

This procedure describes how to configure Control-M/EM Web Server to work with HTTPS, which secures data between the Control-M client applications and the Control-M Web Server using a PKCS#12 keystore.

NOTE: If you are working in a Control-M/EM Distributed environment with multiple Control-M Web Servers, you must provide a different keystore for each server.

NOTE: If you are working in a high availability environment with SSL, run the following command on both the primary and secondary hosts.

```
em restore_host_config -interface_name -name FQDN
```

To configure Control-M/EM Web Server to work with HTTPS:

1. Save the `<keystoreFilename>` in the `%EM_HOME%/ini/ssl` directory.
2. From a command line, type the following:
   ```
   manage_webserver -action create_secure_connection
   ```
   **NOTE:**
   - The keystore filename and the keystore password are prompted after you run the command.
   - The keystore password is encrypted in the Web Server.
3. In the CCM, recycle the Control-M Web Server.
4. Verify that the connector is working and all Control-M clients know the new URL.
5. Delete the http connector as follows:
   a. From a command line, type the following:
      ```
      manage_webserver
      ```
   c. Press the number of the HTTP connector that you want to delete.
      A confirmation message appears.
   d. Press Y.
   e. In the CCM, recycle the Control-M Web Server.
Reverting back to the HTTP connector

This procedure describes how to revert back to an HTTP connector if you deleted it.

➢ To revert back to the HTTP connector:

1. Do one of the following:
   - To replace the `server.xml` file if the delete action was the last action in the manage_webserver utility, do the following:
     a. Navigate to the following directory:
        
        UNIX: %EM_HOME%/etc/emweb/tomcat/conf/
        Windows: %EM_HOME%/emweb/tomcat/conf
     
     b. Delete the `server.xml` file
     
     c. Rename the `server.xml.backup` file to `server.xml`.
   - To edit the `server.xml` file add the http connector:
     
     ```xml
     <Connector port="18080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />
     ```

2. In the CCM, recycle the Web Server.

Enabling SSL between Control-M/EM and Control-M/Server

This procedure describes how to enable SSL between Control-M/EM and Control-M/Server.

➢ To enable SSL:

1. From the Components Tree pane, select the Control-M/EM component and from the Home tab, in the Definitions group, click System Parameters.

   The Control-M/EM System Parameters dialog box appears.

2. In the left pane, click Advanced.

3. From the system parameters table, in the Name column double click the CmsCommMode parameter.

4. The CONTROL-M/EM - Update System Parameter dialog box appears.

5. In the Value field, type AUTO.

6. Click Save.
7. Restart the Control-M Configuration Server to implement the change.

8. Stop Control-M components:
   a. In the Control-M Configuration Manager, stop all components.
   b. Stop the config agent, naming service, and CMS as follows:
      o UNIX: From the Control-M/EM account, run:
         ```
         stop_config_agent
         stop_cms
         stop_ns_daemon
         ```
      o Windows:
         Stop the config agent from the Services window (accessible from the Windows Task Manager).
         From the command line, stop the CMS, by running `stop_cms`.
         Stop the Naming service from the Services window (accessible from the Windows Task Manager).

9. Start the Control-M components:
   • Start the config agent:
     For UNIX, from the Control-M/EM account, run `start_config_agent`.
     For Windows: Start the config agent from the Services window (accessible from the Windows Task Manager).

---

### Enabling SSL between a distributed Control-M/EM and Control-M/Server

This procedure describes how to enable SSL between a distributed Control-M/EM and Control-M/Server.

➢ To enable SSL:

1. From the Components Tree pane, select the Control-M/ EM component and from the Home tab, in the Definitions group, click System Parameters.
   The Control-M/ EM System Parameters dialog box appears.

2. In the left pane, click Advanced.

3. From the system parameters table, in the Name column double click the CmsCommMode parameter.

4. The CONTROL-M/ EM - Update System Parameter dialog box appears.

5. In the Value field, type AUTO.

6. Click Save.

7. Restart the Control-M Configuration Server to implement the change.
8. Stop Control-M components:
   a. In the Control-M Configuration Manager, stop all components.
   b. Stop the config agent, as follows:
      o **UNIX**: From the Control-M/EM account, run:
        
        ```
        stop_config_agent
        ```
      
      o **Windows**: Stop the config agent from the Services window (accessible from the Windows Task Manager).

9. Start the Control-M components:
   - Start the config agent:
     For UNIX, from the Control-M/EM account, run `start_config_agent`.
     For Windows: Start the config agent from the Services window (accessible from the Windows Task Manager).

---

**Enabling SSL on Control-M/Server**

This procedure describes how to enable SSL on the Control-M/Server, using the ctmsys utility. You must repeat this procedure for every Control-M/Server.

➢ **To enable SSL using the ctmsys utility:**

1. Log on to the server computer as the Control-M for Databases owner (for example, user controlm).
2. Run the ctmsys utility.
   
   For more information about the ctmsys utility, see `ctmsys`.

   The following menu is displayed:

   ```
   +------------------------------------------------+ 
   | Control-M SYSTEM MAINTENANCE UTILITY | 
   | Main Menu | 
   +------------------------------------------------+ 
   
   1) Shout Destination Tables 
   2) System Parameters 
   q) Quit 
   ```

3. In the ctmsys main menu, select option 2 **System Parameters**.
4. Enter **n** to move to the next page of parameters.
5. Set option 9 **Secure Sockets Layer** to **ENABLED**.

   **NOTE:** When you set option 9 to **ENABLED**, all agents are automatically set to use SSL.
Enabling SSL on Control-M/Agents

This procedure describes how to enable SSL on Control-M/Agents using the Control-M Configuration Manager.

➢ To modify the settings of each agent according to its required configuration:

1. In Control-M Configuration Manager, right-click the required Control-M/Agent and select Properties.
2. In the Communication tab click the down-arrow next to Secure Socket Layer and select the required value. The values are:
   - Default – inherit the value from the Control-M/Server configuration
   - Enabled – the connection between the Agent and the Control-M/Server is SSL mode irrespective of the Server connection mode
   - Disabled – the connection between the Agent and the Control-M/Server is TCP mode irrespective of the Server connection mode
3. Click Test to check that your settings are correct and workable.
4. Once the test has validated the settings, click OK.

   The connection mode for the agent can be set for any of the valid values. The Control-M server will adjust to the changes made.

   **NOTE:** BMC recommends that switching from SSL Enabled to the server default mode (when the mode is set to DISABLED) must be performed as follows:
   - Set the agent to SSL disabled and then wait for the agent to become available again.
   - When the agent is available (connecting in TCP mode), set the agent to work in default mode.

Enabling SSL between Control-M/EM Server and Control-M add-ons

This procedure describes how to set up SSL communication between Control-M/EM and Control-M add-ons, such as BIM and Control-M Web.

**NOTE:** If Control-M/EM Server and the add-on are on separate machines, you must perform the following steps on both machines.

➢ To configure SSL between Control-M/EM Server and Control-M add-ons:

1. Stop Control-M components:
   a. In the Control-M Configuration Manager, stop all components. For each component, select the component, then right-click and select Desired State > Down.
   b. Stop the config agent, naming service, and CMS as follows:
For UNIX, from the Control-M/EM account, run:

- `stop_config_agent`
- `stop_cms`
- `stop_ns_daemon`

For Windows:

- Stop the config agent from the Services window (accessible from the Windows Task Manager).
- From the command line, stop the CMS, by running `stop_cms`.
- Stop the Naming service from the Services window (accessible from the Windows Task Manager).

2. On UNIX computers only, enter the following command:

```
setenv DISPLAY <terminal_IP_address>
```

3. Start the Domain Configuration (orbconfigure) wizard with one of the following:

- *[UNIX]* `orbconfigure`
- *[Windows]* `orbconfigure.vbs` (under \bin)

The Domain Configuration window is displayed.

4. In the **Domain Settings** panel do the following:

a. Select the **Use Secure Sockets Layer (SSL)** check box.

b. In the given path, under **Use TAO internal configuration file** check box, change the file name `client_server.conf` to `ssl_client_server.conf`.

5. Click **Next**. The Naming Service panel is displayed (If needed configure the Host and Port values).

6. Click **Next**. The summary of the Domain Configuration settings is displayed. Click **Finish**.

7. Run: `orbadmin ns start`.

8. Start the Control-M components:

a. Start the config agent:

   For UNIX, from the Control-M/EM account, run `start_config_agent`.

   For Windows: Start the config agent from the Services window (accessible from the Windows Task Manager).

b. In the Control-M Configuration Manager, start all components. For each component, select the component, then right-click and select **Desired State > Up**.
Enabling SSL between a distributed Control-M/EM Server and Control-M add-ons

This procedure describes how to set up SSL communication between a distributed Control-M/EM server and Control-M add-ons.

To configure SSL between a distributed Control-M/EM Server and Control-M add-ons:

1. Stop Control-M components:
   a. In the Control-M Configuration Manager, stop all components. For each component, select the component, then right-click and select **Desired State > Down**.
   b. Stop the config agent:
      - **UNIX**: From the Control-M/EM account, run: `stop_config_agent`
      - **Windows**: Stop the config agent from the Services window (accessible from the Windows Task Manager).

2. On UNIX computers only, enter the following command:
   `setenv DISPLAY <terminal_IP_address>`

3. Start the Domain Configuration (orbconfigure) wizard with one of the following:
   - * [UNIX] orbconfigure
   - * [Windows] orbconfigure.vbs (under \bin)
   The Domain Configuration window is displayed.

4. In the **Domain Settings** panel do the following:
   a. Select the **Use Secure Sockets Layer (SSL)** check box.
   b. In the given path, under **Use TAO internal configuration file** check box, change the file name `client_server.conf` to `ssl_client_server.conf`.

5. Click **Next**. The Naming Service panel is displayed (If needed configure the Host and Port values).

6. Click **Next**. The summary of the Domain Configuration settings is displayed. Click **Finish**.

7. Run: `orbadmin ns start`.

8. Start the Control-M components:
   a. Start the config agent:
      - **UNIX**: From the Control-M/EM account, run `start_config_agent`.
      - **Windows**: Start the config agent from the Services window (accessible from the Windows Task Manager).
   b. In the Control-M Configuration Manager, start all components. For each component, select the component, then right-click and select **Desired State > Up**.
Disabling SSL between Control-M/EM Server and Control-M client (Changing back to TCP)

This procedure describes how to revert the process of using SSL between Control-M/EM Server and Control-M client.

**NOTE:** If Control-M/EM Server and Control-M client are on separate machines, you must perform the following steps on both machines.

➢ To change back to TCP:

1. Stop Control-M components:
   a. In the Control-M Configuration Manager, stop all components. For each component, select the component, then right-click and select **Desired State > Down**.
   b. Stop the config agent, naming service, and CMS as follows:
      o For UNIX, from the Control-M/EM account, run:
        ```
        stop_config_agent
        stop_cms
        stop_ns_daemon
        ```
      o For Windows:
        Stop the config agent from the Services window (accessible from the Windows Task Manager).
        From the command line, stop the CMS, by running **stop_cms**.
        Stop the Naming service from the Services window (accessible from the Windows Task Manager).

2. On UNIX computers only, enter the following command:
   ```
   setenv DISPLAY terminal_IP_address
   ```

3. Start the Domain Configuration (orbconfigure) wizard with one of the following:
   - * [UNIX] orbconfigure
   - * [Windows] orbconfigure.vbs

4. In the **Domain Settings** panel do the following:
   a. Clear the **Use Secure Sockets Layer (SSL)** check box.
   b. In the given path, under the **Use TAO internal configuration file** check box, change the file name back to `client_server.conf`.

5. Click **Next**. In the **Naming Service** panel do the following:
   a. Click **Show local settings**.
   b. Clear the **Use TAO internal configuration** file check box.

6. Click **Next**. The summary of the Domain Configuration settings is displayed. Click **Finish**.

7. Start all the Control-M/EM components. `orbadmin ns start`
Disabling SSL between a distributed Control-M/EM Server and Control-M client (Changing back to TCP)

This procedure describes how to revert the process of using SSL between a distributed Control-M/EM Server and Control-M client.

➢ To change back to TCP:

1. Stop Control-M components:
   a. In the Control-M Configuration Manager, stop all components. For each component, select the component, then right-click and select **Desired State > Down**.
   b. Stop the config agent:
      o **UNIX**: From the Control-M/EM account, run:
        `stop_config_agent`
      o **Windows**: Stop the config agent from the Services window (accessible from the Windows Task Manager).

2. On UNIX computers only, enter the following command:
   `setenv DISPLAY terminal_IP_address`

3. Start the Domain Configuration (orbconfigure) wizard with one of the following:
   - * [UNIX] `orbconfigure`
   - * [Windows] `orbconfigure.vbs`

4. In the **Domain Settings** panel do the following:
   a. Clear the **Use Secure Sockets Layer (SSL)** check box.
   b. In the given path, under the **Use TAO internal configuration file** check box, change the file name back to `client_server.conf`.

5. Click **Next**. In the **Naming Service** panel do the following:
   a. Click **Show local settings**.
   b. Clear the **Use TAO internal configuration** file check box.

6. Click **Next**. The summary of the Domain Configuration settings is displayed. Click **Finish**.

7. Start all the Control-M/EM components. `orbadmin ns start`

Configuring secure communication between the web server and GUI Server

Apache Tomcat Web server communicates with the Control-M/EM GUI Server with SSL, using the JacORB implementation of CORBA.

You can find SSL parameters for JacORB in the `jacorb.properties` file (`<Control-M/EM_directory>/etc/jacorb.properties`). They are described in SSL parameters for JacORB (on page 21).
**NOTE:** For information on creating a keystore for use with Apache Tomcat Web server web components, see Exporting or importing private/public keys.

➢ To configure Apache Tomcat Web server to work with SSL:

1. In the `jacorb.properties` file, set the `jacorb.security.support_ssl` parameter to **on**.

   Depending on whether you are using a Sun JSSE or IBM JSSE implementation (example IBM AIX), uncomment (delete hashtags) necessary code and use either IBMX509 or SunX509, as shown in the following examples:

   **EXAMPLE: Sun JSSE implementation**
   ```properties
   #Which algorithms to use to initialize the client/server SSL socket factories
   #Use IbmX509 with IBM JDKs
   jacorb.security.jsse.server.key_manager_algorithm=SunX509
   jacorb.security.jsse.server.trust_manager_algorithm=SunX509
   jacorb.security.jsse.client.key_manager_algorithm=SunX509
   jacorb.security.jsse.client.trust_manager_algorithm=SunX509
   ```

   **EXAMPLE: IBM JSSE implementation**
   ```properties
   # Which algorithms to use to initialize the client/server SSL socket factories
   # Use IbmX509 with IBM JDKs
   jacorb.security.jsse.server.key_manager_algorithm=IbmX509
   jacorb.security.jsse.server.trust_manager_algorithm=IbmX509
   jacorb.security.jsse.client.key_manager_algorithm=IbmX509
   jacorb.security.jsse.client.trust_manager_algorithm=IbmX509
   ```

2. In the Control-M Configuration Manager, recycle the Control-M Web Server.

3. Continue with Configuring secure communication between Control-M client applications and the Web Server (on page 11).

➢ To configure Apache Tomcat Web server to work with TCP/IP:

1. Edit the `jacorb.properties` file manually.

2. Set the `jacorb.security.support_ssl parameter` to **off**.

3. In the Control-M Configuration Manager, recycle the Control-M Web Server.
SSL parameters for JacORB

The following table describes SSL parameters for JacORB.

### SSL parameters for JacORB in the jacorb.properties file

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jacorb.security.support_ssl</td>
<td>Indicates whether SSL is enabled. Valid values: on (uses the SSL protocol), off (uses the TCP/IP protocol). Default: off.</td>
</tr>
<tr>
<td>jacorb.security.keystore</td>
<td>Contains the full path and name of the keystore file.</td>
</tr>
<tr>
<td>jacorb.security.keystore_password</td>
<td>Contains the keystore file password.</td>
</tr>
<tr>
<td>jacorb.security.keystore_password_crypt</td>
<td>For encrypting the passphrase. On or off.</td>
</tr>
</tbody>
</table>

### Configuring NamingViewer (browser for Naming Service)

The NamingViewer utility supports the browsing of secure naming services that use SSL with JacORB implementation of CORBA.

You can find SSL parameters for JacORB in the **jacorb.properties** file, in the following directory:

<Control-M/EM_directory>/etc/jacorb.properties

1. **To enable browsing of secure naming services with SSL:**
   1. In the **jacorb.properties** file, set the **jacorb.security.support_ssl** parameter to **on**.
   2. JacORB client on IBM (example IBM AIX) must set the following parameters in **jacorb.properties** file (for IBM JSSE implementation):
      ```
      jacorb.security.jsse.server.key_manager_algorithm=IbmX509
      jacorb.security.jsse.server.trust_manager_algorithm=IbmX509
      jacorb.security.jsse.client.key_manager_algorithm=IbmX509
      jacorb.security.jsse.client.trust_manager_algorithm=IbmX509
      ```
      The default value for all the above parameters above is **SunX509** (Sun JSSE implementation).

2. **To browse non-secure naming services:**
   1. Edit the **jacorb.properties** file manually.
   2. Set the **jacorb.security.support_ssl** parameter to **off**.
Configuring communication with LDAP or Active Directory servers using SSL

The following procedure describes how to enable SSL for LDAP and Active when Control-M/EM is installed on UNIX and Linux operating systems.

➢ To enable SSL:

1. Place the `em_ldap_ssl.pem` file into the `<Control-M/EM_directory>/etc/keystore directory.  
   **NOTE:** If you are working in a high availability environment, place the certificate file (.pem) in the same location in both installations.

2. Verify that a randomness device is installed on the Control-M/EM computer as follows:
   a. Locate either the `random` or `urandom` file in the `/dev` directory. If you find the `random` file, verify that its path is part of the search path.
   b. If neither of these files exist, open the `<Control-M/EM_directory>/etc/ldap.conf` file in a text editor.
   c. Locate the `#TLS_RANDFILE <Control-M/EM_directory>/ini/ssl/rnd.bin` line and remove the `#` character.
   d. Save the modified file.

3. Set an environment variable named "LDAPCONF" with a value pointing to the "ldap.conf" file, which gets set by the EM UNIX account profile.
   **EXAMPLE:**
   ```bash
   setenv LDAPCONF <Control-M/EM_directory>/ctm_em/etc/ldap.conf
   ```

4. Restart all EM components by applying stop_all and start_all commands.

5. Define an LDAP server that can communicate with Control-M/EM in SSL mode, as described in Defining LDAP system parameters.
   If you do not apply all of the above steps, LDAP authentication in SSL mode fails.

6. Recycle the GUI and CMS components.
   See the following example: Obtaining a certificate file from the Windows Active Directory server (on page 22).

Obtaining a certificate file from the Windows Active Directory server

The following procedure provides an example on how to obtain a certificate file from the Windows Active Directory server. The `.pem` format certificate file name must be renamed `em_ldap_ssl.pem`. The rename procedure is outlined in the Active Directory server example in step 8b.

1. Select Programs => Administrative Tools => Certification Authority to open the Certification Authority application.

2. Right-click Certification Authority, and select Properties.

3. Click View Certificate to view the certificate’s page.

4. In the Details tab, click Copy to file to start the Certificate Export Wizard.
5. In the **Export File Format** page, select the Base-64 Encoded X.509 (.cer) format and click **Next**.

6. Enter a file name with a .cer extension that includes the Active Directory server name.

7. Complete the steps in the wizard to create an exported copy of the Certification Authority for the Active Directory server.

8. Convert the certificate from .cer format to .pem format as follows:
   a. Using FTP or another file copying application, copy the Active Directory server certificate file you just created to a system on which the Active Directory client runs.
   b. Log on to the system where you copied the certificate and run the following command:

   ```
   openssl x509 -in AD certificate name -out em_ldap_ssl.pem
   ```

   **NOTE:** For a certificate file obtained from a different LDAP server, rename the file `em_ldap_ssl.pem`.

   The location and name of the certificate (.pem) file can be changed by configuring the TLS_CACERT parameter value in the `<Control-M/EM_directory>/etc/ldap.conf` file for the new path and name.

For **Control-M/ EM installed on Windows**:

9. Obtain a .pem format certificate file from the directory server. Creating and exporting certificate files are different for each LDAP server vendor. Refer to your LDAP server administrator to obtain the correct certificate file. For an example on how to obtain a certificate from the Windows Active Directory, see the Example above.

10. Place the certificate file in the proper location and follow the SSL certificate installation instructions, as provided by Microsoft, using the MMC utility.

For more information about continuing the LDAP and SSL configuration, see Administration.

---

**Deploying SSL on BMC Components**

This procedure describes how to run an SSL deployment script on all relevant Control-M components. Repeat the procedure below for all components, for example: Control-M/EM Servers, Control-M/Server, and Control-M/Agent.

**NOTE:** To update the Control-M/EM certificate, see **Certificates** (on page 25).

➢ **To apply the certificates on Control-M components:**

1. Copy the directory `Certificate_for_<component name>` to a temporary directory in the computer where the component is installed, for example: `<tempLocation>`.

2. Stop the component.

3. From the root directory of the Control-M component, run the following command:

   - **Windows** - `<tempLocation>/setup.bat`
   - **UNIX** - `<tempLocation>/setup.sh`
EXAMPLE: If you placed the deployment directories in /p/Control-M_v6_new_demo, and your Control-M/Enterprise Manager is installed at /bmc/Control-M_EM/Default, then you must run in a UNIX shell:

```$ cd /bmc/Control-M_EM/Default
$ /p/Control-M_v6_new_demo/Certificate_for_CONTROL-M_EnterpriseManagerServers\setup.sh
```

The files are deployed to the required locations and the Control-M component uses either the default keystore password, or if you have specified a Key Store Password, the password by which the Certificates Key Store is locked, is used instead.

4. Restart the relevant component.

Restoring a previous certificate

If you want to automatically restore a previous certificate from a backup for Control-M/EM Client, Control-M/EM Server, Control-M/Server and Control-M/Agent, run the setup script from the backup, as follows:

- **UNIX**: `<sslBackupDir>/setup.sh`
- **Windows**: `<sslBackupDir>\setup.bat`

The setup scripts save a backup of the certificate state prior to the deployment in a separate directory in the `ssl_backup` directory.
Certificates

BMC recommends that you replace the existing certificates using one of the following methods:

- **Generate new certificates using the Control-M Configuration Manager** (on page 25): Use this method when you are using the site CA provided by BMC (different per customer). The site CA is stored in the Control-M/EM Server machine, and is used to sign the certificates for Control-M/EM, Control-M/Server, and Control-M/Agent.

- **Bringing your own certificate** (on page 27) (signed by a different certified CA), then you must be able to provide the certificates, their private key, and the trusted CA.

**NOTE:** To ensure a secure connection between both components, verify both are signed by the same CA.

**NOTE:** In high availability or distributed environments, each Web component needs its own trusted CA that is installed on all Control-M clients as root CA.

Before applying certificates for Control-M/EM for SSL with Control-M/Server, you must verify the following:

- All Control-M/Servers must be version 8.0.00.600 or version 9.0.00.400 or higher.
- For Control-M/Server 8.0.00.600, you must also create or modify the `use_openssl` key string value to `Y` in the registry (Windows) or in the policy file `co.plc` and `ca.plc` (UNIX) for both client and server:
  - **Windows:**
    
    HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\Control-M/Server\SecurityPolicy\CO\server  
    HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\Control-M/Server\SecurityPolicy\CA\server  
  - **UNIX:** In the policy file `co.plc` and `ca.plc`, create or modify the line `"use_openssl=Y"`

Generating new certificates

This procedure describes how to generate new certificates for Control-M/EM and Control-M/Server and replace the pre-installed certificates, which includes updating expiration days and generating the certificates.

**➢ To generate certificates:**

1. In the Control-M Configuration Manager, from the **Home** tab, select **System Configuration > Control-M/EM System Parameters**.

2. Define SSL parameters, as described in Defining SSL system parameters.

3. Select the component in the left pane, and then from the **Tools** drop-down list, select **Security > Manage SSL**.
4. If you want to use the BMC-provided demo certificate as is, select **Use the following site certificate authority**, and continue as follows:
   a. The parameter fields in the first screen are populated with values supplied by BMC. Click **Next**.
   b. Select one of the following:
      a. **All Components of Control-M** to generate certificates for all components
      b. **By Component Type**, and then select the components from the drop-down list.
         You can also enter a **Unique Component Instance ID (email)**. You can do this for all components of this type, or for each instance of this component. Note that this option is not available for the CONTROL-M/EM Server component.
   
   **NOTE:** For CONTROL-M for z/OS only, you can enter a **Key Store Password**, which must be eight characters in length.
   c. Accept the default location to save the generated certificates, or enter a new path.
   d. Click **Next** to generate the certificates, and then **Submit** after the generation process finishes.

5. Create a new and unique instance of the pre-installed site Certificate Authority, select **Create new Certificate Authority for the site**, and then do the following:
   a. Click **Yes** to accept generating a new certificate.
      You are informed that certificates will be generated for all the Control-M components.
   b. Enter the Country Name, Common Name (FQDM), Email Address and other optional parameters of the CA and click **Next**.
   c. If you want to use a password, enter the password and click **Next**.
   d. Accept the default location to save the generated certificates, or type in a new path.
   e. Click **Next** to generate the certificates.
   f. Click **Submit** after the generation process finishes.
      The new certificate deployment directories are created in the location you requested in the CCM client machine.

6. Continue the process by copying the component certificate directories to their relevant locations on the machines with the Control-M components, and by running the deployment scripts on those machines, as described in **Deploying SSL on BMC Components** (on page 23).

### Generating your own certificate

This procedure describes how to generate your own certificate using a BMC provided script. The script `Manage_SSL_Generate_CSR.bat / .sh` is installed on the Control-M/EM Server computer and generates a **KEYPAIR** and a certificate signing request (CSR).
To generate your own certificate:

1. Generate the RSA private key store file as a PEM file in the work area, as follows:

   ```
   Manage_SSL_Generate_CSR.bat/.sh -workarea <workarea path> -genkey [-keysize <size>] -password <password>
   ```

   Where:
   - `-genkey` generates the key pair phase.
   - `-keysize` is 2048 or 4096. Default: 2048
   - `-password` is the password of the private key.

   **EXAMPLE:** `Manage_SSL_Generate_CSR.bat -workarea /tmp/agent1 -genkey -password secret`

2. Generate the CSR, as follows:

   ```
   Manage_SSL_Generate_CSR.bat/.sh -workarea c:\temp\jj -gencsr -subj <text> | -subj_in_workarea -password secret
   ```

   Where:
   - `-gencsr` generates the CSR phase.
   - `-subj` is the X509 Subject and it must be formatted as /type0=value0/type1=value1/type2=...
     Characters may be escaped by \ (backslash) and no spaces are skipped.

   **EXAMPLE:** `"/C=US/ST=Texas/L=Houston/O=BMC Software Inc/OU=Workload Automation/CN=cn/emailAddress=cn@company"

   - `-subj_in_workarea` indicates that the `-subj` parameter input is in the work area already.

   The request.csr file that contains the CSR is now created in the work area.

3. Send the CSR to your Certificate Authority (CA) or your Information Security group (Infosec).

4. Request from your CA or Infosec the following files and copy them to the work area:

   - `ssl certificate <ssl certificate>.pem`
   - `CA certificate .pem` (CA certificate chain)

5. Run the `Manage_SSL_BYO` script to generate a deployment directory for the certificate and private key, as described in *Bringing your own certificate* (on page 27).

**Bringing your own certificate**

This procedure describes how to apply your own certificates signed by approved third-parties. This is achieved by running the `Manage_SSL_BYO` script on the Control-M/EM Server, and then running the script that was generated in the deployment directories, on the relevant components' installations.

**NOTE:** To update the Control-M/EM certificate, see *Certificates* (on page 25).

The certificate that you apply must be a valid X509 certificate.
The following components are supported:

- Control-M Agent
- Control-M Server
- Control-M Enterprise Manager Servers
- Control-M for Web Services, Java and Messaging
- Control-M Web
- Control-M Enterprise Manager API
- Control-M zOS

Before you begin

You must already have one or all of the following: the signed certificate, its private key, and its CA chain in one of the following formats:

- PEM (Privacy Enhanced Mail) files for each of the following: Certificate, Private Key, and Trusted Root CAs
- PKCS#12 file with the Certificate, private key, and CAs. If the PKCS#12 contains multiple certificates and key pairs, then you must also have available the name of the pair to use.

**NOTE:** All components must be signed using the same CA Certificate. Private key passwords must exist for PKCS#12 and PEM, however the Certificate PEM must not be locked/protected by password.

➢ To apply the certificates:

1. Create a temporary folder in the root directory of the computer where the Control-M/EM resides.
2. In the directories of the deployed components, add the scripts to import your SSL certificates and keys. See Examples of script usage when importing certificates and keys (on page 28).

Examples of script usage when importing certificates and keys

Examples of scripts for importing your SSL certificates and keys are presented below.

**PEM format:** Manage_SSL_BYO -input pem

- component {component name}
- output { Manage_SSL output deployment directory}
- output_keystores_password {Component keystore password}
- certificate {certificate pem file}
- private_key {certificate private key pem file}
- password {password for private key pem file}
- ca_certificates {CA certificate chain PEM file}

**PKCS#12 format:** Manage_SSL_BYO -input PKCS#12
-component {component name}
-output { Manage_SSL output deployment directory}
-output_keystores_password {Component keystore password}

-file {pkcs#12 file to import from}
-password {password of pkcs#12 file}

[-cert_and_key_name {name of the certificate and private key to import }]

Supported -component values:

CONTROL-M_Agent
CONTROL-M_Server
CONTROL-M_EnterpriseManagerServers
CONTROL-M_BPI
CONTROL-M_Web_Application
CONTROL-M_EnterpriseManagerAPI
CONTROL-M_zOS

EXAMPLE: PEM file usage in an environment using three agents:

Manage_SSL_BYO -input -component Control-M_Agent -output /tmp/Agent1_SSL_Deployment -output_keystores_password abcd1234 -input pem -certificate agent1cert.pem -private_key agent1key.pem -password secret -ca_certificates CAs.pem

Manage_SSL_BYO -input -component Control-M_Agent -output /tmp/Agent2_SSL_Deployment -output_keystores_password abcd1234 -input pem -certificate agent2cert.pem -private_key agent2key.pem -password secret -ca_certificates CAs.pem

Manage_SSL_BYO -input -component Control-M_Agent -output /tmp/Agent3_SSL_Deployment -output_keystores_password abcd1234 -input pem -certificate agent2cert.pem -private_key agent3key.pem -password secret -ca_certificates CAs.pem

EXAMPLE: PKCS#12 usage

For PKCS#12 with a single certificate and key pair (cert_and_key_name parameter not required):

Manage_SSL_BYO -input -component Control-M_Agent -output /tmp/Agent2_SSL_Deployment -output_keystores_password abcd1234 -input PKCS#12 -password 1234abcd -file /p12files/ag.p12

For PKCS#12 with multiple certificates and key pairs:

Manage_SSL_BYO -input -component Control-M_Agent -output /tmp/Agent2_SSL_Deployment -output_keystores_password abcd1234 -input PKCS#12 -password 1234abcd -file /p12files/All.p12 -cert_and_key_name ag
Certificate copies for Control-M for z/OS

For Control-M for Z/OS you need to pass the directory content to your z/OS Control-M administrator.

The following table describes the keystore files for z/OS.

<table>
<thead>
<tr>
<th>Key store file</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOAGATE.p12</td>
<td>Export the certificate for Control-M for Z/OS with the key-pair to be used by IOAGATE in PKCS#12 format. The password for the PKCS#12 file is displayed in the summary window that is generated when running the Generate Component Certificates wizard.</td>
</tr>
<tr>
<td>CA.pem</td>
<td>Export the certificate of the Site CA that signed the certificate in PEM format when CLIAUTH=YES (which uses client authentication) is defined in IOAGATE.</td>
</tr>
</tbody>
</table>

For more information about how to use these files, see the INCONTROL for z/OS Installation Guide, Appendix B "IOAGATE installation and configuration considerations, SSL support".
Advanced Settings and Configurations

The following sections include additional information and details about security settings and configurations:

- Configuring a Control-M/Agent to use SSL (on page 34)
- Changing a server-agent connection mode for an existing agent (on page 34)
- Configuring Control-M/EM communication with Control-M/Server to use SSL (managed Control-M instances) (on page 35)
- Configuring Control-M/EM API JacORB (on page 41)
- Setting up a signed certificate for a Java keystore (on page 43)
- Configuring security policies (on page 44)

SSL communication parameters

This section briefly describes the SSL-related parameters that determine communication modes for Control-M/Server, Control-M/Agent, and Control-M/EM.

Control-M/Server

The Secure Sockets Layer system parameter determines the communication mode the Server uses to communicate with Agents and Control-M/EM. You can set this parameter to any of the communication modes shown in the following table.

**Control-M/Server communication modes**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENABLED</td>
<td>Control-M/Server works in SSL mode. When attempting to connect to an Agent that is in SSL=N mode (discussed subsequently), the Server tries to switch the Agent to SSL=Y mode.</td>
</tr>
<tr>
<td>INACTIVE</td>
<td>Control-M/Server works in non-SSL mode. When attempting to connect to an Agent that is in SSL=Y mode, the Server tries to switch the Agent to SSL=N mode.</td>
</tr>
<tr>
<td>DISABLED</td>
<td>Control-M/Server works in non-SSL mode. When attempting to connect to an Agent that is in SSL=Y mode, the Server does not try to switch the Agent to SSL=N mode.</td>
</tr>
</tbody>
</table>

**NOTE:** Changing the Server communication mode from ENABLED to DISABLED can cause all agents to become unavailable. In that case, you need to change the Server mode to INACTIVE and wait until all required agents are available again. Then, you can change the Server mode to ENABLED or DISABLED.
**Control-M/Agent**

For Control-M/Agent, the COMMOPT parameter determines the Agent’s communication mode. Valid values for COMMOPT are SSL=Y (communication is enabled) or SSL=N (communication is disabled).

- On Microsoft Windows computers, COMMOPT is in the Control-M/Agent registry.
- On UNIX computers, COMMOPT is in the agent_home/ctm/data/CONFIG.dat file.

**Key Store files**

This section describes the Key Store files that are used by Control-M.

<table>
<thead>
<tr>
<th>Key Store file</th>
<th>Control-M component</th>
</tr>
</thead>
</table>
| **KDB**  
key database file | Control-M/Agent 9.0.00.200 and lower  
Control-M/Server 9.0.00.300 and lower  
Control-M/EM servers (Gateway, CMS) 9.0.00.400 and lower |
| **PEM**  
Privacy enhanced mail | Control-M/EM servers (GSR and BIM server)  
Control-M/EM client |
| **Java KeyStore** | Control-M/Server 9.0.00.300 and lower  
Control-M/EM EMAPI client  
Control-M Web Services and Messaging API  
Control-M/EM BMC Batch Impact Manager |
| **PKCS#12** | Control-M/EM servers (Gateway, CMS) 9.0.00.500 and higher  
Control-M/Agent 9.0.00.300 and higher  
Control-M/Server 9.0.00.400 and higher  
Control-M for z/OS |

**NOTE:** For background information about SSL, see SSL documentation on the Internet. For more information about Control-M authentication and privacy levels, see Configuring security policies (on page 44).
Deploying the PKCS#12 file to Control-M components

This procedure describes how to deploy the PKCS#12 input file to all Control-M components using the ctmkeytool utility.

➢ To deploy the PKCS#12 file:

1. Run the ctmkeytool utility from the full path for Control-M/EM, Control-M/Server, and Control-M/Agent:
   - **Control-M/EM**: `<EM_Home>\bin\`
   - **Control-M/Server**: `<Server_Home>\ctm_server\scripts\`
   - **Control-M/Agent**: `<Agent_Home>\exe\`

2. From a command line, type the following:
   ```
   ctmkeytool -keystore <name and path of PKCS12 key store file> -password <key store password> -passwkey <name and path of password key file>
   ```

Generating a PKCS#12 keystore

This procedure describes how to generate a PKCS#12 keystore for a certificate using openSSL.

➢ To generate a public-private key pair:

1. Generate a private key, as follows:
   ```
   openssl genrsa -passout pass: <password> -aes256 -out key_pass.pem 2048
   ```

2. Remove a passphrase from the key, as follows:
   ```
   openssl rsa -passin pass:<password> -in key_pass.pem -out key_nopass.pem
   ```

3. Generate a CSR, as follows:
   ```
   openssl req -new -sha256 -key key_nopass.pem -out req.csr -subj "/emailAddress=ctm_security@bmc.com/CN=CTM_certs/C=US/O=BMC/OU=MPM/L=Houston/ST=Houston"
   ```

4. Send the following request to your CA:
   ```
   openssl x509 -req -days 7300 -in req.csr -out cert.pem -sha256 -passin pass:<password> -CAcreateserial -CA key cacert.pem -CA cacert.pem
   ```

5. Create the PKCS#12 file with signed certificate from your CA (cert.pem), as follows:
   ```
   ```
Configuring a Control-M/Agent to use SSL

For each Control-M/Agent on which you want to configure SSL, complete the appropriate procedure:

<table>
<thead>
<tr>
<th>For</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control-M/Agent for UNIX</td>
<td>In the \agent_home\ctm\data\CONFIG.dat file, set \COM\MOPT to SSL=Y.</td>
</tr>
<tr>
<td>Control-M/Agent for Microsoft Windows (version 6.4.01 and later)</td>
<td>Run the ctmagcfg utility, select option 7 (Advanced Parameters), and specify Y for option 8 in the Advanced menu.</td>
</tr>
<tr>
<td>Control-M/Agent for Microsoft Windows (versions earlier than 6.4.01)</td>
<td>Run the ctmagcfg utility, and specify Y for option 16 (SSL).</td>
</tr>
</tbody>
</table>

**NOTE:** Completing this step can save time if you have a large number of agents that work with Control-M/Server. If you skip this step, Control-M/Server automatically makes a one-time request to set the SSL parameter. This request requires between two and five minutes for each agent.

To configure a new agent, you can use Control-M Configuration Manager or ctm_menu. You can set one or more Agents to SSL mode and other Agents to TCP mode. For example, you can use Control-M/Server to work with the majority of the agents it is connected to in SSL mode, and can connect to other agents in TCP mode.

When adding a Control-M/Agent to a Control-M/Server using Control-M Configuration Manager to configure the Control-M/Agent to work with SSL, click the down-arrow next to the Secure Socket Layer field. The values are:

- **Default** - inherit the value from the Control-M/Server configuration
- **Enabled** - the connection between the Agent and Control-M Server is in SSL mode, irrespective of the Server connection mode
- **Disabled** - the connection between the Agent and Control-M Server is in TCP mode irrespective of the Server connection mode

Changing a server-agent connection mode for an existing agent

This procedure describes how to modify the settings of each agent according to its required configuration.

➢ To change server-agent connection mode:

1. In Control-M Configuration Manager, right-click the required Control-M/Agent, and select Properties.
2. In the Communication tab, click the down-arrow next to Secure Socket Layer and select the required value. The values are:
Control-M SSL Guide

- **Default** - inherit the value from the Control-M/Server configuration
- **Enabled** - the connection between the Agent and the Control-M/Server is SSL mode irrespective of the Server connection mode
- **Disabled** - the connection between the Agent and the Control-M/Server is TCP mode irrespective of the Server connection mode

3. Click **Test** to check that your settings are correct and workable.

4. Once the test has validated the settings, click **OK**.

The connection mode for the agent can be set for any of the valid values. The server will adjust to the changes made.

**NOTE:** BMC recommends that switching from SSL Enabled to the server default mode (when the mode is set to DISABLED) must be performed in the following steps:

- Set the agent to SSL disabled and then wait for the agent to become available again.
- When the agent is available (connecting in TCP mode), set the agent to work in default mode.

### Configuring Control-M/EM communication with Control-M/Server to use SSL (managed Control-M instances)

This procedure describes how to configure Control-M/EM communication with Control-M/Server to use SSL (managed Control-M instances).

- **To configure Control-M/EM communication with Control-M/Server to use SSL (managed Control-M instances):**
  1. Set the value of CMS parameters to **auto**.
  2. Restart the Control-M Configuration Server to implement the change.

### Configuring Control-M/EM communication with Control-M/Server to use SSL (unmanaged Control-M instances)

The following procedure describes how to configure Control-M/EM communication with Control-M/Server to use SSL (unmanaged Control-M instances).

- **To configure SSL for unmanaged Control-M instances:**
  1. Log in to Control-M Configuration Manager.
  2. Use the left panel of the Configuration Manager window to select a Server definition:
     a. At the bottom of the panel, select the **By Computer** tab.
     b. Expand the **Control-M/Server** node of the **All Components** tree.
c. Select the Control-M/Server definition you want to configure.
   
   The components of the selected definition are displayed in the right panel of the window.

3. Double-click the line displaying the Control-M/Server definition component you want to configure.
   
   The Control-M Definition window is displayed.

4. In the Protocol field of the definition window, select SSL_ENABLE or TCP, and click OK.

5. Use the Control-M Configuration Manager to stop and restart the Control-M/EM Gateway to implement the change.
   
   For more information about the Control-M Configuration Manager, see Administration.

At startup, the Gateway tries to communicate with the Server using TCP/IP protocol. If the Server does not respond during the synchronization interval (90 seconds by default), the Gateway automatically changes its protocol to SSL and tries to communicate by using the SSL protocol.

Storing certificates for TAO

The following procedure describes storing the CA and signed certificates.

Default CA and application certificates are provided and stored in standard PEM format.

➢ To store a Root Certificate of Authority (CA) and signed certificates:

1. Place the certificates (ca_cert.pem, cert_name.pem, and cert_name_priv_key.pem) in the `<Control-M/EM_directory>/ini/ssl/new_ca.pem` directory.

2. Update the ssl_client_server.conf and ssl_ns.conf files in the `<Control-M/EM_directory>/etc` directory by changing the names of the demonstration certificates to the names of your certificates.

   Parameters for ssl_client_server.conf are explained in ssl_client_server.conf parameters (on page 37).

   EXAMPLE: If the original content of the ssl_client_server.conf file is:

   ```
   dynamic SSLIOP_Factory Service_Object *
   TAO_SLIOP:__make_TAO_SLIOP_Protocol.Factory() "
   -SSLAuthenticate SERVER_AND_CLIENT
   -SSLPri
   ```

   Change the full path name of the certificates (bold above) to the names of your certificates.

   In this example, authentication of both the server and the client is required because the -SSLAuthenticate parameter is set to SERVER_AND_CLIENT.
ssl_client_server.conf parameters

The following table describes parameters in the `ssl_client_server.conf` file.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-SSLAuthenticate</td>
<td>Indicates whether authentication is required for server, client, or both. Valid values: SERVER, CLIENT, SERVER_AND_CLIENT</td>
</tr>
<tr>
<td>-SSLPPrivateKey</td>
<td>Points to the location of the private key.</td>
</tr>
<tr>
<td>-SSLCertificate</td>
<td>Points to the location of the public key.</td>
</tr>
<tr>
<td>-SSLCAfile</td>
<td>Points to the CA certificate. Default: <code>&lt;Control-M/EM_directory&gt;/ini/ssl/new_ca.pem</code></td>
</tr>
<tr>
<td></td>
<td>The CA certificate, public key, and private key files can be replaced.</td>
</tr>
<tr>
<td>-SSLrand</td>
<td>Points to a binary file used to generate random numbers for dynamically encrypting communications between client and server. The file provided by Control-M/EM can be replaced with another binary file. Client and server binary files are independent and do not need to match. Default: <code>&lt;Control-M/EM_directory&gt;/ini/ssl/rnd.bin</code></td>
</tr>
</tbody>
</table>

**NOTE:** This parameter is optional for Windows installations.

Private key password

The private key password for demonstration certificates is stored in the `ClientServerSSL.ini` file in the `<Control-M/EM_directory>/ini/ssl` directory. Control-M/EM components read and decode this password and provide it to the SSL layer.

➢ To update the private key password for use with your site’s certificates:

1. Go to the `<Control-M/EM_directory>/ini/ssl` directory.
2. Update the `ClientServerSSL.ini` file with the new encrypted password by entering the command `cryptocli new_password ClientServerSSL.ini`

Naming service certificate

The Naming Service requires the private key password interactively during startup. This requirement prevents users from activating the Naming Service in batch mode.

In the Control-M/EM demonstration certificates, the password has been stripped from the private key so that the Naming Service can be invoked without entering the password. The `ssl_ns.conf` configuration file points to the stripped private key file.

➢ To activate the Naming Service using a new stripped private key:

1. Use the `ssl_ns.conf` file for the Naming Service.
2. Place the stripped private key file in the `<Control-M/EM_directory>/ini/ssl` directory.

3. Update the `ssl_ns.conf` file with the new private key file name, as described in 2 referred to in Storing certificates for TAO (on page 36) for the `ssl_client_server.conf` file.

➢ To activate the Naming Service interactively using a secure private key:

In the Naming Service panel, set TAO internal configuration file to the same file that Control-M/EM CORBA servers and clients use: `<Control-M/EM_directory>/etc/ssl_client_server.conf`

However, this alternative requires that the PEM password be entered interactively, and therefore the Naming Service cannot be run as a Windows service.

Certificate expiration

Control-M/EM comes with demonstration SSL certificates with an expiration period of 4 years. The client applications check certificate expiration on each connection attempt. The client issues a warning if the certificate expires in less than the number of days specified in the `WarningSSLExpirationDays` system parameter, as described in General parameters. Valid values: 1 - 365. Default: 60.

If an SSL certificate is going to expire in less than the number of days specified in this parameter, a message is displayed in the Message column of the Control-M Configuration Manager main window, and a record is written to the application log.

Common SSL error messages

The following are Common SSL error messages:

**Message 1**

ACE_SSL (2372 | 1656) error code: 336151576 - error:14094418:SSL routines:SSL3_READ_BYTES:tlsvl alert unknown ca
Failed to register in the CORBA services.

*Explanation:* The GUI server fails to resolve a secure Naming Service. The `-SSLCAfile` parameter is not specified in the TAO configuration file, or it points to an invalid location.

*Corrective Action:* Determine which reason caused the failure and correct the problem.

**Message 2**

ACE_SSL (3632|2580) error code: 336134278 - error:14090086:SSL routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
Failed to register in the CORBA services.

*Explanation:* The GUI server fails to resolve a secure Naming Service for one of the following reasons:

The `-SSLCAfile` parameter is not specified in the TAO configuration file.

The `-SSLCAfile` parameter points to an invalid location.

The CA PEM file (*new_ca.pem*) is corrupted.

The CA PEM file (*new_ca.pem*) doesn't match the certificates used.

*Corrective Action:* Determine which reason caused the failure and correct the problem.
Message 3
Failed to register in the CORBA services.
Explanation: An attempt was made to connect to a non-secure Naming Service.
Corrective Action: Ensure that the attempted connection is to a secure Naming Service and ensure that the Naming Service was started as a secure Naming Service.

Message 4
ClientServerSSL.ini was not found at D:\ Program Files\ BMC Software\ Control-M EM 7.0.00\ Default\ ini\ ssl
dynamic initialization failed for SSLIOP_Factory
(3868|2956) Unable to initialize the Service Configurator: Invalid argument
Failed to register in the CORBA services.
Explanation: The GUI server fails to initialize. The ClientServerSSL.ini file was not found in the directory.
Corrective Action: Ensure that the ClientServerSSL.ini file is located in the directory.

Message 5
Password decryption error. Key string file may be corrupted.
Unknown error dynamic initialization failed for SSLIOP_Factory
(1556|2364) Unable to initialize the Service Configurator: Invalid argument
Failed to register in the CORBA services.
Explanation: The GUI server fails to initialize. The ClientServerSSL.ini file is corrupted or contains a password that was encrypted using the wrong key.
Corrective Action: Ensure that the ClientServerSSL.ini file is not corrupted and contains a properly encrypted password.

Message 6
dynamic initialization failed for SSLIOP_Factory
(3868|3820) Unable to initialize the Service Configurator: Invalid argument
Failed to register in the CORBA services.
Explanation: The GUI server fails to initialize. The private or public key certificate was not found.
User Response: Ensure that the -SSLPrivateKey parameter points to the file containing the private key. Ensure that the -SSLCertificate parameter points to the file containing the public key. When using the demonstration certificates, the default values are:
-SSLPrivateKey 'PEM:/home/ctm_em/ini/ssl/CertDemoU_pk.pem'
-SSLCertificate'PEM:/home/ctm_em/ini/ssl/CertDemoU.pem"

Message 7
TAO (2196|3224) Service Configurator unable to open file be D:\ Program Files\ BMC Software\ Control-M EM 7.0.00\ Default\ ini\ ssl
(2196|3224) Unable to initialize the Service Configurator: Invalid argument
Failed to register in the CORBA services.
Explanation: The GUI server fails to initialize. The configuration file referenced in the `-ORBSvcConf` parameter was not found. For more information, see the example in Storing certificates for TAO (on page 36).

Corrective Action: Ensure that the file that is being pointed to exists in the specified location.

CORBA::TRANSIENT exceptions

Why do I get a CORBA::TRANSIENT exception when using SSLIOP?

A CORBA::TRANSIENT exception usually indicates that the client was unable to connect to the server when attempting to invoke a request. For standard IIOP, this normally occurs when the client cannot resolve the hostname embedded in the IOR or cannot reach the specified IP address.

In the case of SSLIOP, a CORBA::TRANSIENT exception may also be thrown when the certificates in use are invalid (for example, expired), or the certificate authority certificate has not been set.

Configuring NamingViewer (browser for Naming Service)

The NamingViewer utility supports the browsing of secure naming services that use SSL with JacORB implementation of CORBA.

You can find SSL parameters for JacORB in the `jacorb.properties` file, in the following directory:

`:Control-M/EM_directory>/etc/jacorb.properties`

- To enable browsing of secure naming services with SSL:
  1. In the `jacorb.properties` file, set the `jacorb.security.support_ssl` parameter to `on`.
  2. JacORB client on IBM (example IBM AIX) must set the following parameters in `jacorb.properties` file (for IBM JSSE implementation):
     - `jacorb.security.jsse.server.key_manager_algorithm=IbmX509`
     - `jacorb.security.jsse.server.trust_manager_algorithm=IbmX509`
     - `jacorb.security.jsse.client.key_manager_algorithm=IbmX509`
     - `jacorb.security.jsse.client.trust_manager_algorithm=IbmX509`

The default value for all the above parameters above is `SunX509` (Sun JSSE implementation).

- To browse non-secure naming services:
  1. Edit the `jacorb.properties` file manually.
  2. Set the `jacorb.security.support_ssl` parameter to `off`. 
Using your own encrypted password

The keystore password for demonstration certificates is not encrypted. To use an encrypted password, run the changePass utility as follows:

- (UNIX) `changePass` in the `<Control-M/EM_directory>/bin` directory
- (Windows) `changePass` in the `<Control-M/EM_directory>\bin` directory

This utility accepts a keytool password, encrypts it, and updates the `jacorb.security.keystore` and `jacorb.security.keystore_password_crypt` parameters in the `jacorb.properties` file.

**NOTE:** If you configure the `<Control-M/EM_directory>/etc/jacorb.properties` file to use SSL, you will not be able to browse non-secure naming services.

Configuring Control-M/EM API JacORB

The SSL parameters for JacORB are in the `jacorb.properties` file.

These parameters are described in [JacORB parameters](#) (on page 42).

➢ To configure Control-M/EM APIs to use the SSL protocol:

1. Run `emapi-configure` with the `-ssl` option, or edit the `jacorb.properties` file manually as follows:
   a. Set the `jacorb.security.support_ssl` parameter to `on`.
   b. Set the `ORBInitRef.NameService` parameter to `corbaloc:ssl/op:ns_host:ns_port/NameService` (replace `ns_host` and `ns_port` with the correct values).

   For information about `emapi-configure`, see Control-M/EM API Installation.

2. JacORB client on IBM (example IBM AIX) must set the following parameters in `jacorb.properties` file (for IBM JSSE implementation):

   - `jacorb.security.jsse.server.key_manager_algorithm=IbmX509`
   - `jacorb.security.jsse.server.trust_manager_algorithm=IbmX509`
   - `jacorb.security.jsse.client.key_manager_algorithm=IbmX509`
   - `jacorb.security.jsse.client.trust_manager_algorithm=IbmX509`

   The default value for all the parameters above is `SunX509` (Sun JSSE implementation).

   The `jacorb.properties` file is located in the following directory:
   `<EM_API>/etc/jacorb.properties`

➢ To configure Control-M/EM APIs to use the TCP/IP protocol:

- Run `emapi-configure` without the `-ssl` option, or edit the `jacorb.properties` file manually as follows:
  a. Set the `jacorb.security.support_ssl` parameter to `off`.
  b. Set the `ORBInitRef.NameService` parameter to `corbaloc:iiop:ns_host:ns_port/NameService` (replace `ns_host` and `ns_port` with the correct values).
JacORB parameters

The following table describes JacORB parameters.

### SSL parameters for JacORB in the jacorb.properties file

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jacorb.security.support_ssl</td>
<td>Indicates whether SSL is enabled. Valid values: on (use the SSL protocol), off (use the TCP/IP protocol). Default: off.</td>
</tr>
<tr>
<td>jacorb.security.keystore</td>
<td>Contains the full path and name of the keystore file.</td>
</tr>
<tr>
<td>jacorb.security.keystore_password</td>
<td>Contains the keystore file password.</td>
</tr>
<tr>
<td>jacorb.security.keystore_password_crypt</td>
<td>Indicates whether the keystore file password is encrypted. Valid values: on (yes), off (no). Default: off.</td>
</tr>
</tbody>
</table>

Processing SSL certificates with JacORB

The application is provided with a default CA certificate and default application certificates in key database (keystore) format for use with JacORB.

The default parameter values for the demonstration certificates are:

- jacorb.security.keystore=emapi_root/ etc/ keystore/ emapi.keystore
- jacorb.security.keystore_password=emdemo
- jacorb.security.keystore_password_crypt=off

These parameters are in the jacorb.properties file.

NOTE: For more information on certificates, see Processing certificates and Certificate expiration (on page 38)

Setting up a signed certificate for a non-Java Key Database (KDB)

In order to work in the Control-M/EM Server environment, you must configure both a Java keystore and a non-Java keystore.

This procedure explains how you configure a non-Java keystore.

Setting up a signed certificate for a non-Java keystore includes the following procedures:
Control-M SSL Guide

- Installing a trusted root authority certificate
- Generating public-private key pairs
- Creating and installing the signed certificate
- Creating a certificate signing request
- Installing the signed certificate

When you're done with the steps above, proceed with Setting up a signed certificate for a Java keystore (on page 43).

Locations of alias IDs for public-private key pairs

The following default alias names are specified in the UNIX .plc file or the Microsoft Windows Registry:

<table>
<thead>
<tr>
<th>For communication from</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control-M/Server to Control-M/Agent</td>
<td>NSDN is specified in the ns.plc file.</td>
</tr>
<tr>
<td>Control-M/Server to Control-M/EM</td>
<td>CODN is specified in the co.plc file.</td>
</tr>
<tr>
<td>Control-M/Server Configuration Agent to Control-M Configuration Server</td>
<td>CADN is specified in the ca.plc file.</td>
</tr>
<tr>
<td>Control-M/Agent to Control-M/Server</td>
<td>AGDN is specified in the ag.plc file.</td>
</tr>
<tr>
<td>Control-M/EM to Control-M/Server</td>
<td>CODN is specified in the gtw.plc file.</td>
</tr>
<tr>
<td>Control-M/EM to Control-M Configuration Agent</td>
<td>CADN is specified in the cmsg.plc file.</td>
</tr>
</tbody>
</table>

Setting up a signed certificate for a Java keystore

In order to work in the Control-M/EM Server environment, you must configure both a Java keystore and a non-Java keystore.

This procedure explains how you configure a Java keystore.

Before proceeding, you must first start with Setting up a signed certificate for a non-Java Key Database (KDB) (on page 42).

Setting up a signed certificate for a Java keystore includes the following procedures:

- Generating public-private key pairs (on page 44)
- Creating a certificate signing request (on page 44)
- Installing the signed certificate (on page 44)
- Installing the signed certificate (on page 44)
Generating public-private key pairs

A cryptographic key pair is a set of two cryptographic keys (one public and one private) that is used to start an SSL session.

Before requesting a certificate from the CA, you must use this procedure to generate a cryptographic key pair and assign that key pair to the new certificate.

```
keytool -genkey -alias CEDN -keyalg RSA -keystore ctmkey.jks -keysize (number of bits for example, 1024)
```

Creating a certificate signing request

This procedure describes how to create a signing request.

Follow the on-screen prompts.

```
keytool -certreq -v -alias CEDN -file req.csr -keystore ctmkey.jks
```

Installing a trusted root authority certificate

To use SSL, you must obtain a trusted root authority certificate (CA) from an organization that validates digital certificates used for online transactions. A certificate is validated by a hierarchy of CAs that approve the certificate. The ultimate CA in the chain is the trusted root certificate authority.

```
keytool -importcert -trustcacerts -alias CA -file cacert.pem -keystore ctmkey.jks -storepass abcd1234 -noprompt (Import CA into JKS)
```

Installing the signed certificate

```
keytool -import -v -alias CEDN -file cert.pem -keystore ctmkey.jks (Import signed certificate into JKS)
```

Configuring security policies

The security policy is defined by entries made in security policy tables. A Site Policy table is required for each major Control-M component (Control-M/Server, Control-M/Agent, and Control-M/EM). The entries in these Site Policy tables provide the basic framework for the Control-M site's security policy. Additions and modifications to the Site Policy, if needed, are defined in optional Application Policy tables for various Control-M functions. Entries in these tables add to and supersede the entries in the Site Policy tables.

On UNIX computers, the security policy tables are contained in .plc files. On Microsoft Windows computers, these tables are contained in the Registry.

SSL communication policy is based on "variable value" pairs – called attributes – that are stored in Policy Tables. Each UNIX stanza (or Microsoft Windows Registry key) contains appropriate attributes. Some attributes do not apply to certain functions, some do not apply to certain security levels, and some cannot be changed.
Security policy is implemented by assigning values to the attribute variables described in the Security policy table referred to in Security policy variables (on page 54). Default policy values for each major Control-M component are specified in that component’s site.plc file or site Registry hive.

When a network communication connection is established, the profile for that connection is obtained from variables in the .plc files (for UNIX) or in the Registry (for Microsoft Windows). The .plc files are described on Sample .plc files (on page 46). The Microsoft Windows Registry is described on Microsoft Windows environment (on page 48).

Changes to the key database, key database password, and security policy do not take effect until you restart Control-M/Server, Control-M/Agent, and Control-M/EM.

**UNIX environment**

In the UNIX environment, Policy Tables are implemented in ASCII text Policy Files in standard .INI format. Policy Tables are stored in .plc files located in these directories:

- `<CONTROL-M/Server_directory>/ctm_server/data/SSL/cert`
- `<CONTROL-M/Agent_directory>/ctm_server/data/SSL/cert`
- or `<CONTROL-M/Server_directory>/ctm_agent/ctm/data/SSL/cert`
- `<CONTROL-M/EM_directory>/etc/site/resource/SSL/cert`

Stanzas in the Site Policy and Application Policy files specify the security module that supports the role defined by the stanza. If an application acts like a network server, security attributes are obtained from the [server] stanza. If an application acts like a network client, security attributes are obtained from the [client] stanza.

A typical Site Policy is shown in Control-M/Server site.plc file (on page 46). When establishing the type of communication listed in the table below, the values (if any) in the relevant application .plc file override the values in the site.plc file.

<table>
<thead>
<tr>
<th>Application .plc file</th>
<th>Type of communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>ns.plc</td>
<td>Control-M/Server to Control-M/Agent</td>
</tr>
<tr>
<td>co.plc</td>
<td>Control-M/Server to Control-M/EM</td>
</tr>
<tr>
<td>ca.plc</td>
<td>Control-M/Server Configuration Agent to Control-M Configuration Manager</td>
</tr>
<tr>
<td>ag.plc</td>
<td>Control-M/Agent to Control-M/Server</td>
</tr>
<tr>
<td>gtw.plc</td>
<td>Control-M/EM gateway to Control-M/Server</td>
</tr>
<tr>
<td>cmsg.plc</td>
<td>Control-M Configuration Server to Control-M Configuration Agent</td>
</tr>
<tr>
<td>em.plc</td>
<td>(for Control-M/EM internal encryption purposes)</td>
</tr>
</tbody>
</table>
Sample .plc files

Sample .plc files similar to the following are provided with the installation:

- Control-M/Server co.plc file (on page 46)
- Control-M/Server site.plc file (on page 46)
- Control-M/Server ns.plc file (on page 47)
- Control-M/Enterprise Manager site.plc file (on page 47)
- Control-M/Enterprise Manager site.plc file (on page 47)

Control-M/Server co.plc file

[server]
identity=CODN
logfile=cosrv.log

[client]
logfile=cocln.log
identity=CODN
keyfile=$CONTROLM/data/SSL/cert/ctmkey.kdb

Control-M/Server site.plc file

[server]
bindir=<CONTROLM>/exe_<MACHINE>
bindir64=<CONTROLM>/exe_<MACHINE>
keyfile=ctmkey.kdb
security_level=4
logdir=$CONTROLM/data/SSL/log
loglevel=ERROR,WARNING,INFO,TRACE
securitydir=$CONTROLM/data/SSL/cert
sk8dir=$CONTROLM/data/SSL/cert
password=a877b993b0b40c558176bb07efc54da43505b61b5d07d9d,<CONTROLM>/data
/SSL_directory/cert/tree.bin

[client]
bindir=<CONTROLM>/exe_<MACHINE>
bindir64=<CONTROLM>/exe_<MACHINE>
keyfile=ctmkey.kdb
security_level=4
logdir=$CONTROLM/data/SSL/log
loglevel=ERROR,WARNING,INFO,TRACE
securitydir=$CONTROLM/data/SSL/cert
sksdir=$CONTROLM/data/SSL/cert
password=a877b993b0b40c558176bbb07efc54da43505b61b5d07d9d,<CONTROLM>/data/SSL_directory/cert/tree.bin

Control-M/Server ns.plc file

[server]
identity=NSDN
logfile=nssrv.log
security_level=3

[client]
identity=NSDN
logfile=nscln.log
keyfile=$CONTROLM/data/SSL/cert/ctmkey.kdb

Control-M/Enterprise Manager site.plc file

[client]
bindir=$EM_HOME/appl/lib/bin.$ARCH
bindir64=$EM_HOME/appl/lib/bin.$ARCH
keyfile=gtwkey.kdb
security_level=4
logdir=$EM_HOME/site/resource/ssl/log
loglevel=ERROR
securitydir=$EM_HOME/site/resource/ssl/cert
sksdir=$EM_HOME/site/resource/ssl/cert

Control-M/Enterprise Manager co.plc file

[client]
logfile=gtw_ssl.log
identity=CODN
keyfile=$EM_HOME/site/resource/ssl/cert/gtwkey.kdb
password=a877b993b0b40c558176bbb07efc54da43505b61b5d07d9d,$EM_HOME/site/resource
Microsoft Windows environment

A security policy is defined by string entries in Policy Tables in the Windows Registry. The Site Policy key is always required. Its path is:

"\HKEY_LOCAL_MACHINE\SOFTWARE\Bmc Software\CONTROL-M/Server\SecurityPolicy\site"

**NOTE:** BMC does not recommend editing the Windows registry unless you have experience working with the registry and you back up the registry before proceeding.

The basic security policy is defined by Site Policy keys. Modifications, if needed, are defined by optional Application Policy keys.

Policy Tables contain string entries that specify the security module that supports the function defined by the keys in the Windows Registry. The communications security policy is determined by the role the application is playing: client or server. Therefore, the Policy Tables contain two communications keys, one for the server function:

"\HKEY_LOCAL_MACHINE\SOFTWARE\Bmc Software\CONTROL-M/Server\SecurityPolicy\site\server"

and one for the client function:

"\HKEY_LOCAL_MACHINE\SOFTWARE\Bmc Software\CONTROL-M/Server\SecurityPolicy\site\client".

Sample Policy Tables for Microsoft Windows are listed under Control-M/Server registry (on page 49). The Policy Tables are at the following Registry location:

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC\
Software\CONTROL-M\{Agent|Server}\SecurityPolicy\{site|NS|CA|CO|AG}\{client|server|common}]

Values (if any) specified in the relevant NS, CA, CO, and AG. Registry keys override the values specified in the site Registry key.

Sample Policy Tables for Microsoft Windows are listed under Control-M/Enterprise Manager registry (on page 52). The Policy Tables are at the following Registry location:

HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\9.0.X\SecurityPolicy\{site|GTW|CMSG|EM\}{client|server|common}

Values (if any) specified in the relevant GTW, CMSG, and EM. Registry keys override the values specified in the site Registry key.

**NOTE:** The EM Registry contains an EM key for internal encryption purposes. Do not change this key.

To set the TLS1.2 SSL Protocol, see Setting the TLS1.2 SSL Protocol (on page 49).
Setting the TLS1.2 SSL Protocol

This procedure describes how to set the TLS1.2 SSL Protocol in the Policy table.

To set the TLS1.2 SSL Protocol:

1. Navigate to the Policy table in Windows Registry, as described in Microsoft Windows environment (on page 48).

2. Add the following **provider_options** key in the registry:
   - **Value name:** provider_options
   - **Value data:** SSLProtocol=TLS1_2,[TLSCipherSuite=]

   **EXAMPLE:** Value name: provider_options
   - Value data: SSLProtocol=TLS1_2,TLSCipherSuite=DHE-RSA-AES256-SHA256

   For the TLSCipher list, see Cipher list (on page 55).

3. After updating the configuration, you need to restart the component.

   You can update the site hive in the registry to affect all communications or update a specific hive to control a specific connection.

Sample Microsoft Windows registry keys

These are the following Default Registry key entries for Control-M/Server and Control-M/EM:

- Control-M/Server registry (on page 49)
- Control-M/Enterprise Manager registry (on page 52)

Control-M/Server registry

REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M/Server\SecurityPolicy]

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M/Server\SecurityPolicy\CO]

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M/Server\SecurityPolicy\CO\client]
"logfile"="cocln.log"
"keyfile"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\data\ssl\cert\ctmkey.kdb"
"identity"="CODN"
[HKEY_LOCAL_MACHINE\SOFTWARE\BMC
  Software\CONTROL-M/Server\SecurityPolicy\CO\server]
  "identity"="CODN"
  "logfile"="cosrv.log"

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC
  Software\CONTROL-M/Server\SecurityPolicy\NS]}

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC
  Software\CONTROL-M/Server\SecurityPolicy\NS\client]
  "identity"="NSDN"
  "logfile"="nscln.log"
  "keyfile"="D:\Program Files\BMC Software\CONTROL-M
  Server\ctm_server\data\ssl\cert\ctmkey.kdb"

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC
  Software\CONTROL-M/Server\SecurityPolicy\NS\server]
  "identity"="NSDN"
  "logfile"="nssrv.log"
  "security_level"="3"

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC
  Software\CONTROL-M/Server\SecurityPolicy\site]}

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC
  Software\CONTROL-M/Server\SecurityPolicy\site\client]
  "bindir"="D:\Program Files\BMC Software\CONTROL-M
  Server\ctm_server\exe"
  "securitydir"="D:\Program Files\BMC Software\CONTROL-M
  Server\ctm_server\data\ssl\cert"
  "logdir"="D:\Program Files\BMC Software\CONTROL-M
  Server\ctm_server\data\ssl\log"
  "loglevel"="ERROR"
  "keyfile"="ctmkey.kdb"
  "security_level"="4"
  "sksdir"="D:\Program Files\BMC Software\CONTROL-M
  Server\ctm_server\data\ssl\cert"
"password"="a877b993b0b40c558176bb07efc54da43505b61b5d07d9d,D:\Program Files\BMC Software\CONTROL-M Server\CTM_SERVER\DATA\SSL\Cert\tree.bin"

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\Server\SecurityPolicy\site\common]
"sksdir"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\data\ssl\cert"
"bindir"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\exe"
"keyfile"="ctmkey.kdb"
"security_level"="4"
"logdir"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\data\ssl\log"
"loglevel"="ERROR, WARNING, INFO, TRACE"
"securitydir"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\data\ssl\cert"
"password"="a877b993b0b40c558176bb07efc54da43505b61b5d07d9d,D:\Program Files\BMC Software\CONTROL-M Server\CTM_SERVER\DATA\SSL\Cert\tree.bin"

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\Server\SecurityPolicy\site\server]
"bindir"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\exe"
"keyfile"="ctmkey.kdb"
"security_level"="4"
"logdir"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\data\ssl\log"
"loglevel"="ERROR"
"securitydir"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\data\ssl\cert"
"sksdir"="D:\Program Files\BMC Software\CONTROL-M Server\ctm_server\data\ssl\cert"
"password"="a877b993b0b40c558176bb07efc54da43505b61b5d07d9d,D:\Program Files\BMC Software\CONTROL-M Server\CTM_SERVER\DATA\SSL\Cert\tree.bin"
Control-M/Enterprise Manager registry

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy]

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\CMG]

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\CMG\client]
"securitydir"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Gtwgcs\appl\ecs\resource\ssl\cert"
"loglevel"="ERROR"
"logfile"="cmgsssl.log"
"keyfile"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Gtwgcs\appl\ecs\resource\ssl\cert\cmgskey.kdb"
"password"="a877b993b0b40c558176b7ff5e54da43505b61b5d07d9d,D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Ini\local\tree.bin"
"identity"="CADN"
"security_level"="4"
"sksdir"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Gtwgcs\appl\ecs\resource\ssl\cert"

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\EM]

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\EM\client]
"securitydir"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Gtwgcs\appl\ecs\resource\ssl\cert"
"identity"="CODN"
"logfile"="emssl.log"
"loglevel"="ERROR"
"password"="a877b993b0b40c558176b7ff5e54da43505b61b5d07d9d,D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Ini\local\tree.bin"
"keyfile"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Ini\local\emkey.kdb"
[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\GTW]

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\GTW\client]
"sksdir"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Gtwgcs\appl\ecs\resource\ssl\cert"
"security_level"="4"
"identity"="CODN"
"password"="a877b993b0b40c558176b8b07efc54da43505b61b5d07d9d,D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Ini\local\tree.bin"
"keyfile"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Gtwgcs\appl\ecs\resource\ssl\cert\gtwkey.kdb"
"logfile"="gtwssl.log"
"loglevel"="ERROR"
"securitydir"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Gtwgcs\appl\ecs\resource\ssl\cert"

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\site]

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\site\client]
"bindir"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\bin"
"logdir"="D:\Program Files\BMC Software\CONTROL-M EM 7.0.00\Default\Ini\local\log"

[HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\CONTROL-M\CONTROL-M/Enterprise Manager\7.0.X\Default\SecurityPolicy\site\common]
@=""
Security policy variables

The following security policy variables are located in the appropriate Microsoft Windows Registry keys and UNIX .plc files.

### Security policy variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>security_level</td>
<td>A digit from 1 through 4. These levels are described in Security levels (on page 56).</td>
</tr>
<tr>
<td>bindir</td>
<td>Absolute path to a subdirectory containing the dynamically loaded security binary modules. For example: C:\Program Files\BMC Software\Control-M Server\ctm_server\exe</td>
</tr>
<tr>
<td>bindir64</td>
<td>bindir for 64-bit computer. For example: C:\Program Files\BMC Software\Control-M Server\exe_MACHINE</td>
</tr>
<tr>
<td>sksdir</td>
<td>Absolute path to a Security Key Store read/write subdirectory where Control-M encrypted keys are stored. For example: C:\Program Files\BMC Software\Control-M Server\etc\site\resource\ssl\cert</td>
</tr>
<tr>
<td>securitydir</td>
<td>Absolute path to read only subdirectory where *.kdb key databases and key material files are stored. For example: &quot;securitydir&quot;=&quot;C:\Program Files\BMC Software\Control-M Server\ctm_server\data\SSL\cert&quot;</td>
</tr>
<tr>
<td>password</td>
<td>Encrypted password (generated by the bmcryptpw utility), followed by a comma, followed by the absolute path of the keymaterial file (used for 3 DES key computation). Embedded blanks are not allowed. See To create an SSL key database (Recommended task summary) and To change the key database password (Maintaining certificates).</td>
</tr>
<tr>
<td>keyfile</td>
<td>Absolute path of the key database file. For example: keyfile=C:\Program Files\BMC Software\Control-M Server\data\SSL_directory\cert\ctmkey.kdb</td>
</tr>
<tr>
<td>identity</td>
<td>Key pair label (CADN, CODN, NSDN, or AGDN) in a key database.</td>
</tr>
<tr>
<td>logdir</td>
<td>Absolute path to subdirectory containing the log file. For example: &quot;logdir&quot;=&quot;C:\Program Files\BMC Software\Control-M Server\ctm_server\etc\site\resource\ssl\log&quot;</td>
</tr>
<tr>
<td>loglevel</td>
<td>One or more of the following values separated by commas: ERROR WARNING INFO TRACE</td>
</tr>
</tbody>
</table>
### Variable Description

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| logfile        | Log file (path and) name. For example: 

```
logfile=gtw_ssl.log
```

| provider_options | provider_options=SSLProtocol=SSLv3,SSLV3CipherSuite =<ciphers list>
provider_options=SSLProtocol=TLS1,SSLV3CipherSuite =<ciphers list>
provider_options=SSLProtocol=TLS1_2,TLSCipherSuite =<ciphers list>

**NOTE:** All ciphers with SHA256 can be used only with TLS1_2 protocol.

---

### Cipher list

The following ciphers can be used:

- AES128-SHA
- AES256-SHA
- AES128-SHA256
- AES256-SHA256
- DES-CBC-SHA
- DES-CBC3-SHA
- DHE-RSA-AES128-SHA
- DHE-RSA-AES256-SHA
- DHE-RSA-AES128-SHA256
- DHE-RSA-AES256-SHA256
- EDH-RSA-DES-CBC-SHA

To use more than one cipher, use a space to separate them.

**EXAMPLE:**

For SSLv3 protocol:
```
provider_options=SSLProtocol=SSLv3,SSLV3CipherSuite=DHE-RSA-AES256-SHA AES256-SHA DHE-RSA-AES128-SHA
```

For TLS1 protocol:
```
provider_options=SSLProtocol=TLS1,SSLV3CipherSuite=DHE-RSA-AES256-SHA AES256-SHA DHE-RSA-AES128-SHA
```

For TLS1_2 protocol:
```
provider_options=SSLProtocol=TLS1_2,TLSCipherSuite=DHE-RSA-AES256-SHA AES256-SHA DHE-RSA-AES128-SHA
```
NOTE: If you work on a Control-M/Server installed on AIX and you plan to use Advanced Encryption Standard (AES) ciphers, you must configure the system to work with the TLS1 SSL protocol.

NOTE: If you plan to configure Control-M/Server to use the SSLv3 protocol with ciphers DHE-RSA-AES128-SHA AES128-SHA or DHE-RSA-AES256-SHA AES256-SHA, the Control-M/Agents must be version 8.0.00.300 and above.

Security levels

For Control-M/Server and Control-M/Agent, the default security level is 3 in ‘server role’ and 4 in ‘client role’. For Control-M/EM gateways, the default security level is always 4.

You must specify the same security level for a pair of components that communicate with each other. There is one exception: you can specify level 3 for communication in ‘server role’ and level 4 for communication in ‘client role’ to the communication channel between Control-M/Agent and Control-M/Server.

Security level 1

Security level 1 provides privacy only. After a secure connection is established, user data is encrypted using TripleDES. This level does not provide authentication. When a client-server connection is established, a session key is generated and exchanged using the Diffie-Helman secure key exchange method. Security level 1 usually prevents access by a casual network browser. To prevent access by a skilled and determined intruder, use security level 2 or later.

Security Level 2

Security level 2 implements the Secure Socket Layer protocol. A server operating at security level 2 accesses a private database of key pairs and retrieves the key pair named in the identity attribute of its security policy. It uses the key pair values and accompanying certificate to establish an SSL connection with the client.

A client operating at security level 2 accepts the server’s certificate. SSL ordinarily requires the client to establish a chain of trust for the server’s certificate down to a trusted root. But, in security level 2, the client omits this step and accepts the server’s certificate if the certificate’s attributes (for example, inception and expiration date) are acceptable.

When using security level 2, the server and client cannot be sure of each other’s identity. Nevertheless, a secure exchange of the session key occurs and privacy superior to that of security level 1 is provided.

Security Level 3

Security level 3 operates like security level 2 except that the client must use its own database of certificates to establish a chain of trust for the server’s certificate down to a trusted root. This is in addition to the requirement that all other attributes of the server’s certificate be acceptable. Therefore, the client can be certain of the identity of the server, but the server cannot be certain of the identity of the client. This connection is said to have server authentication only.
Security Level 4

Security level 4 provides privacy and authentication for both client and server. Security level 4 is enforced by the server. After a handshake with the client as described in security level 3, the server sends a message to the client demanding a rehandshake.

The client returns its own certificate, which the server verifies down to a trusted root. If the client does not provide a certificate that the server can verify, the server shuts down the connection. Since each peer has identified itself to the other, this connection is said to have mutual authentication.

After changing the security level, stop and restart the services listed in the following table to implement the change.

Services to be stopped and restarted

<table>
<thead>
<tr>
<th>Service</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control-M/Server</td>
<td>Introduction to Control-M Configuration Manager</td>
</tr>
<tr>
<td>Control-M/Agent</td>
<td>Introduction to Control-M Configuration Manager</td>
</tr>
<tr>
<td>Control-M/EM</td>
<td>Use the Control-M Configuration Facility to stop and restart Control-M/EM Gateway to implement the change. This facility is described in the Introduction to Control-M Configuration Manager.</td>
</tr>
</tbody>
</table>

Access files

Access files use email fields in server certificates for authentication. Access files can be defined for Control-M/Server and Control-M/Agent. The default access file contains lines similar to these:

```plaintext
[SSL_SERVER]
;
ALLOW_ACL = *
DENY_ACL =
```

The following table describes the parameters in the access file.

Access file parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL_SERVER</td>
<td>Authentication confirming a server’s identity</td>
</tr>
<tr>
<td>ALLOW_ACL</td>
<td>Allows signed certificates to be sent to specified addresses. Default: * (Allow every client).</td>
</tr>
<tr>
<td>DENY_ACL</td>
<td>Deny the sending of signed certificates to specified e-mail addresses. Default: blank (Does not deny any client).</td>
</tr>
</tbody>
</table>

The security level must be 4. For more information, see Security Level 4 (on page 57).
The server certificate email field is checked after the regular SSL handshake, and after both peers have checked that the certificates that they received are signed by a trusted root CA.

DENY_ACL and ALLOW_ACL are used to control the sending of signed certificates to email destinations. For more information, see table Distinguished name information (Creating a certificate signing request).

Include the following lines in an access file to accept only the certificates issued to controlm@bmc.com and email@bmc.com. The access file must deny all other certificates, including those signed by a trusted root.

```
[SSL_SERVER]
;
  ALLOW_ACL = controlm@bmc.com,email@bmc.com
  DENY_ACL =
```