Installing and Configuring AES CTI Engine

Release 4.5.1
Issue 1
September 2016

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya may generally make available to users of its products and Hosted Services. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original Published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked websites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on Avaya hardware and software. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this product while under warranty is available to Avaya customers and other parties through the Avaya Support website: http://support.avaya.com or such successor site as designated by Avaya. Please note that if you acquired the product(s) from an authorized Avaya Channel

Partner outside of the United States and Canada, the warranty is provided to you by said Avaya Channel Partner and not by Avaya.

Hosted Service

THE FOLLOWING APPLIES IF YOU PURCHASE A HOSTED SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE), THE TERMS OF USE FOR HOSTED SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/ LICENSEINFO UNDER THE LINK “Avaya Terms of Use for Hosted Services” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES

THE HOSTED SERVICE. BY ACCESSING OR USING THE HOSTED SERVICE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE DOING SO (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THE TERMS OF USE. IF YOU ARE ACCEPTING THE TERMS OF USE ON BEHALF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO

BIND SUCH ENTITY TO THESE TERMS OF USE. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO ACCEPT THESE TERMS OF USE, YOU MUST NOT ACCESS OR USE THE HOSTED SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE HOSTED SERVICE. YOUR USE OF THE HOSTED SERVICE SHALL BE LIMITED BY THE NUMBER AND TYPE OF LICENSES PURCHASED UNDER YOUR CONTRACT FOR THE HOSTED SERVICE, PROVIDED, HOWEVER,

THAT FOR CERTAIN HOSTED SERVICES IF APPLICABLE, YOU MAY HAVE THE OPPORTUNITY TO USE FLEX LICENSES, WHICH WILL BE INVOICED ACCORDING TO ACTUAL USAGE ABOVE THE CONTRACT LICENSE LEVEL. CONTACT AVAYA OR AVAYA’S CHANNEL PARTNER FOR MORE INFORMATION ABOUT THE LICENSES FOR THE APPLICABLE HOSTED SERVICE, THE AVAILABILITY OF ANY FLEX LICENSES (IF APPLICABLE), PRICING AND BILLING INFORMATION,

AND OTHER IMPORTANT INFORMATION REGARDING THE HOSTED SERVICE.

Support Tools:

“AVAYA SUPPORT TOOLS” MEAN THOSE SUPPORT TOOLS PROVIDED TO PARTNERS OR CUSTOMERS IN CONNECTION WITH MAINTENANCE SUPPORT OF AVAYA EQUYIPMENT (E.G., SAL, SLA MON, AVAYA DIAGNOISTIC SERVER, ETC.) AVAYA SUPPORT TOOLS ARE INTENDED TO BE USED FOR LAWFUL DIAGNOSTIC AND NETWORK INTEGRITY PURPOSES ONLY. The customer is responsible for understanding and complying with applicable legal requirements with regard to its network. The Tools may contain diagnostic capabilities that allow Avaya, authorized Avaya partners, and authorized customer administrators to capture packets, run diagnostics, capture key strokes and information from endpoints including contact lists, and remotely control and monitor end-user devices. The customer is responsible for enabling these diagnostic capabilities, for ensuring users are aware of activities or potential activities and for compliance with any legal requirements with respect to use of the Tools and diagnostic capabilities on its network, including, without limitation, compliance with laws regarding notifications regarding capture of personal data and call recording.

Licenses

THE SOFTWARE LICENSE TERMS OR SUPPORT TOOLS LICENSE TERMS AVAILABLE ON THE AVAYAWEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO

OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AVAYA CHANNEL PARTNER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA CHANNEL PARTNER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE

AVAYA AFFILIATE (“AVAYA”).

Avaya grants you a license within the scope of the license types described below, with the exception of Heritage Nortel Software and Support Tools, for which the scope of the license is detailed below Where the order documentation does not expressly identify a license type, the applicable license will be a Designated System License. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the documentation

or other materials available to you. “Designated Processor” means a single stand-alone computing device. “Server” means a Designated Processor that hosts a software application to be accessed by multiple users.

License type(s)

Designated System(s) License (DS). End User may install and use each copy or an Instance of the Software only on a number of Designated Processors up to the number indicated in the order. Avaya may require the Designated Processor(s) to be identified in the order by type, serial number, feature key, Instance, location or other specific designation, or to be provided by End User to Avaya through electronic means established by Avaya specifically for this purpose.

Concurrent User License (CU). End User may install and use the Software on multiple Designated Processors or one or more Servers, so long as only the licensed number of Units are accessing and using the Software at any given time. A “Unit” means the unit on which Avaya, at its sole discretion, bases the pricing of its licenses and can be, without limitation, an agent, port or user, an e-mail or voice mail account in the name of a person or corporate function (e.g., webmaster or helpdesk), or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software. Units may be linked to a specific, identified Server or an Instance of the Software.

Database License (DL). End User may install and use each copy or an Instance of the Software on one Server or on multiple Servers provided that each of the Servers on which the Software is installed communicates with no more than an Instance of the same database.

CPU License (CP). End User may install and use each copy or Instance of the Software on a number of Servers up to the number indicated in the order provided that the performance capacity of the Server(s) does not exceed the performance capacity specified for the Software. End User may not re-install or operate the Software on Server(s) with a larger performance capacity without Avaya’s prior consent and payment of an upgrade fee.

Named User License (NU). You may: (i) install and use the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use the Software on a Server so long as only authorized Named Users access and use the Software. “Named User”, means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya’s sole discretion, a “Named User” may be, without limitation, designated by name, corporate function (e.g., webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.

Shrinkwrap License (SR). You may install and use the Software in accordance with the terms and

conditions of the applicable license agreements, such as “shrinkwrap” or “clickthrough” license accompanying or applicable to the Software (“Shrinkwrap License”).

Heritage Nortel Software

"Heritage Nortel Software" means the software that was acquired by Avaya as part of its purchase of the Nortel Enterprise Solutions Business in December 2009. The Heritage Nortel Software currently available for license from Avaya is the software contained within the list of Heritage Nortel Products located at http://support.avaya.com/LicenseInfo/ under the link "Heritage Nortel Products", or such successor site as designated by Avaya. For Heritage Nortel Software, Avaya grants Customer a license to use Heritage

Nortel Software provided hereunder solely to the extent of the authorized activation or authorized usage level, solely for the purpose specified in the Documentation, and solely as embedded in, for execution on, or (in the event the applicable Documentation permits installation on non-Avaya equipment) for communication with Avaya equipment. Charges for Heritage Nortel Software may be based on extent of activation or use authorized as specified in an order or invoice.

Support Tools: Avaya Support Tools are provided as an entitlement of Avaya Support Coverage (e.g., maintenance) and the entitlements are established by Avaya. The scope of the license for each Tool is described in its License terms and/or the applicable service description document.

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Hosted Service, or hardware provided by Avaya. All content on this site, the documentation, Hosted Service, and the Product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may

not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Third Party Components

"Third Party Components" mean certain software programs or portions thereof included in the Software or Hosted Service may contain software (including open source software) distributed under third party agreements ("Third Party Components"), which contain terms regarding the rights to use certain portions of the Software ("Third Party Terms"). As required, information regarding distributed Linux OS source code (for those Products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the Documentation or on Avaya's website at: http://support.avaya.com/Copyright or such successor site as designated by Avaya. You agree to the Third Party Terms for any such Third Party Components.

THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE HTTP://WWW.MPEGLA.COM.

Note to Service Provider

The Product or Hosted Service may use Third Party Components subject to Third Party Terms that do not allow hosting and require a Service Provider to be independently licensed for such purpose. It is your responsibility to obtain such licensing.

Preventing Toll Fraud

"Toll Fraud" is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of Toll Fraud associated with your system and that, if Toll Fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Toll Fraud intervention

If you suspect that you are being victimized by Toll Fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support website: http://support.avaya.com or such successor site as designated by Avaya. Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

Trademarks

The trademarks, logos and service marks ("Marks") displayed in this site, the Documentation, Hosted Service(s), and Product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support website: http://support.avaya.com, or such successor site as designated by Avaya.

Contact Avaya Support

See the Avaya Support website: http://support.avaya.com for Product or Hosted Service notices and articles, or to report a problem with your Avaya Product or Hosted Service. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: http://support.avaya.com (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support.

Table of Contents

Chapter 1: Introduction

Purpose

This document describes how to install and configure Avaya Aura® Application Enablement Services (AES) Computer Telephony Integration (CTI) Engine. This document is intended for users who want to install and configure AES CTI Engine.

Change History

Document Version	Date	Author	Summary of Changes	Reviewer
1.0	23-March-2016	Y. Abbas	Initial version	D. Castañares, I. Bedascarrasbure
2.0	1-April-2016	I. Bedascarrasbure	Reviewed version	D. Castañares, D. Iguchi
3.0	15-June-2016	Y. Abbas	Reviewed version	D. Castañares, I. Bedascarrasbure
4.0	28-July-2016	M. West	Updates	
4.1	08-Sept-2016	M. West	Update to 4.5.0	
4.2	13-Sept-2016	M. West	Update to 4.5.1	S. Fernández

Chapter 2: AES CTI Engine Overview

Overview

The CTI Engine is an application that provides a web services interface into the Avaya Telephony Server Application Programming Interface (TSAPI) as provided by Avaya's Application Enablement Services (AES). The specific web services interface that is provided is an XML based message set transported over web sockets.

The interface provided by the CTI Engine is then used by client applications to communicate with the AES and perform CTI activity.

Prerequisites

- Configure the Avaya Application Enablement Server (AES)
- Install Red Hat Enterprise Linux on the server.
- Install additional Linux packages.
- Configure the TSAPI link to the Communication Manager in AES.

  Note

  AES CTI Engine uses application-specific licensing and does not use TSAPI user licenses.

Environment Configuration

Server Requirements

- Intel E520 Quad Core / 2.4GHz processor or superior
- 4 GB of RAM
- 10 GB of free hard disk space
- Red Hat Enterprise Linux (RHEL) 6 (Update 5 and later)

  Note

  The newer versions of RHEL 6 are supported, but ensure that libraries required to support the solution are also installed. AES CTI Engine is also works on a 64-bit version of RHEL, but the libraries required must be 32-bit.

  The required 32-bit libraries are:

  - **`openssl-1.0.1e-42.el6.i686`**

- **expat-2.0.1-11.el6_2.i686**
- **libstdc++-4.4.7-16.el6.i686**
- **libcurl-7.19.7-46.el6.i686**
- **glib2-2.28.8-4.el6.i686**
- **GConf2-2.28.0-6.el6.i686**

The AES CTI Engine makes use of web socket technology between the user's browser and the host server. The communication path between those two entities must permit the use of web sockets.

Communication Manager Requirements

The CM should be configured to have **ASAI Proprietary Feature** enabled. The current setting can be determined using command **display system-parameters customer-options**. Go to Page 9, and look for the heading **ASAI PROPRIETARY FEATURES**. There is a single entry under this heading called **Proprietary?** This setting should be set to **y** for the CTI Engine to work correctly.

Port Requirements

The application uses the following ports for the given products:

Port number	Description	Direction
450	AES TSAPI Port	Out to AES
1066 to1081	TSAPI encrypted links	Out to AES
8090	CTI Engine	In from Client
9090	Client Logger	In from Client

It is important to make sure that these ports are opened on the Linux server in the appropriate direction. Assuming a default RHEL 6.5 installation, the outbound ports will be automatically enabled, but the inbound ports will be blocked. These ports must be opened for the CTI Engine to function correctly.

Preparing the Local Firewall

The firewall program on Linux is **`iptables`**. By default, all ports are open for output, but the input list is very restricted. Use the **`iptables`** command to open access to the ports that are needed. Run the following commands as a `root` user.

```
sudo iptables -I INPUT 2 -p tcp -m tcp --dport 8090 -j ACCEPT
sudo iptables -I INPUT 3 -p tcp -m tcp --dport 9090 -j ACCEPT
```

Save the changes in the **/etc/sysconfig/iptables** folder using the following command as a `root` user:

```
sudo service iptables save
```

To clear the local firewall settings for initial testing, use the following command as a `root` user:

```
sudo iptables --flush
```

Note that some corporate security policies do not allow for iptables to be stopped or empty. Be sure they validate that this is an acceptable option before trying to flush the iptables configuration. Regardless, this is an unacceptable solution for a production system.

Chapter 3: CTI Engine Installation and Configuration

Installing the TSAPI Client

Procedure

1. Run the following command with admin privileges from the location where you placed the installation file:

   ```
   chmod +x tsapi-client-linux-7.0.0-131.bin
   sudo ./tsapi-client-linux-7.0.0-131.bin
   ```

2. Edit the newly installed file **`/usr/lib/tslibrc`**.
   This is the file that specifies the location of the AES server.

3. Replace the line specifying **`127.0.0.1`** with either the IP address or the FQDN of the AES server. If there are two AES servers, enter values for both of them.
   This file must be modified using root privileges. Each AES server must be specified on its own line.

4. Once the AES server is specified run the program **`/usr/lib/tstest`** to make sure the TSAPI client is properly installed.
   If you get a list of TLinks, then the TSAPI client is installed properly. If you get no TLinks, then the AES servers were not properly specified, or the AES servers are not running or available.

5. (Optional) You can also provide the details in the **`/usr/lib/tstest`** program, such as an AES CT User username and password, source extension, and a destination number to fully validate the installation by making a very brief test call.
   When performing the test, getting any kind of “ACS” error means that the link is not working correctly, or the username/password are incorrect. Getting a “CSTA” error means that the origination number was not valid, or there was some other kind of operational error, but that the connection was good. So, even if the test results in a “CSTA” error, the test can be considered successful.

   Note

   The TSAPI client also creates `avaya` as a username that will be used for all steps and processes that do not need root privileges. You should log in as `root` and set a password for `avaya` user.

   There is a feature of Linux called **selinux** that can hinder the operation of the TSAPI client. Ensure that **selinux** is either completely disabled or running in a permissive mode. If **selinux** is not disabled, the TSAPI client will not work properly.

Installing the CTI Engine Application

Procedure

Run the following command from the location where you have placed the executable file:

```
sudo rpm -ivh AES-Connector-CTIEngine-4.5.1-<rev>.i686.rpm
```

Once the CTI Engine application has been installed, make sure that the `avaya` user can make changes to the root directory of the application. To perform this action, run the following command:

```
sudo chown –R avaya:susers /opt/avaya
```

Licensing

The CTI Engine uses user-based licensing. The CTI Engine can use one of the two following modes for licensing: file-based licensing or WebLM licensing.

File-Based Licensing

The license file is created based on the primary MAC address of the user's computer, the number of concurrent users licensed, and whether the call center functionality is enabled.

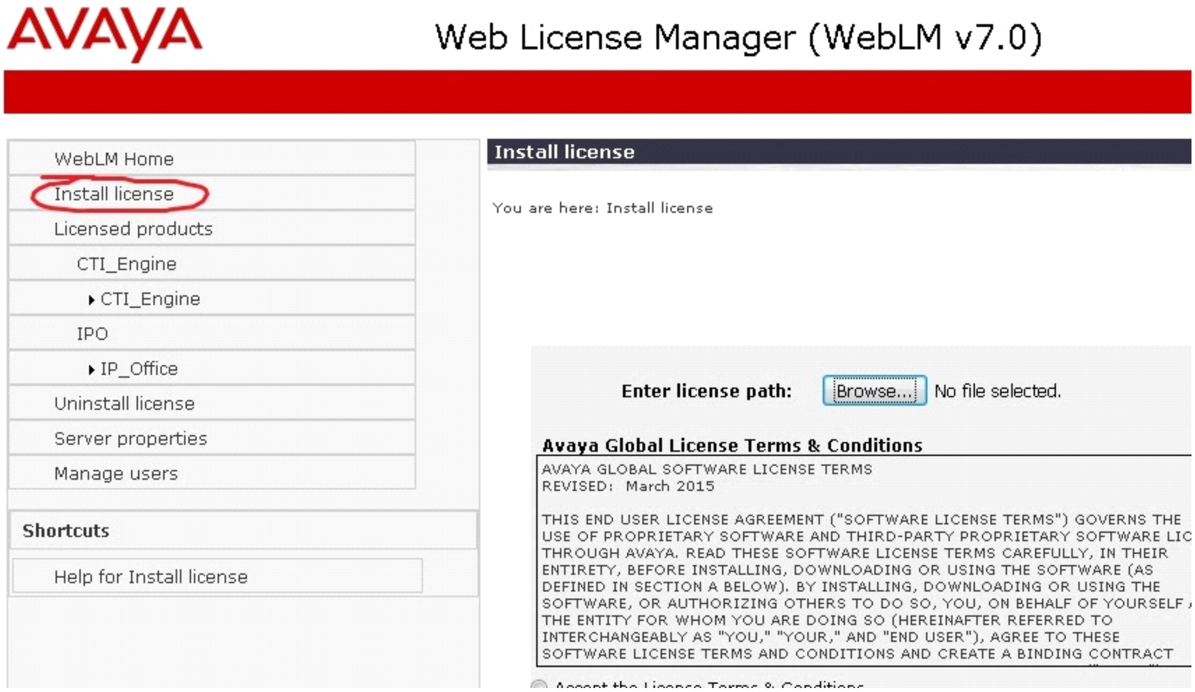
The license file name is **`ctienginelicense`** and must be placed at the location **`/opt/avaya/ctiengine`**. Additionally, you must place the public key file **`license_public_key.pem`** at the location **`/opt/avaya/ctiengine/certificates`**. The license and public key files is sent out once the MAC address of the server is determined.

If the license file does not exist, the CTI Engine will use the WebLM licensing method.

WebLM Licensing

Starting in CTI Engine version 4.5.0, the CTI Engine can use Avaya WebLM server to manage its licensing. In order to use WebLM, the **`ctienginelicense`** file must not be present in the **`/opt/avaya/ctiengine`** directory. WebLM is Avaya's standard tool used for managing Avaya softphone product licenses. Licenses are loaded into the WebLM, and the CTI Engine will communicate with the WebLM server to obtain those licenses. To use those licenses, the CTI Engine will be configured with where to access the WebLM server, and how many licenses it will need to cache.

The licenses are loaded into the WebLM by installing the **`license_cml`** file. The license file is installed using the WebLM's web application interface, and selecting "Install license" button.



AVAYA Web License Manager (WebLM v7.0)

Install license

You are here: Install license

Enter license path: Browse... No file selected.

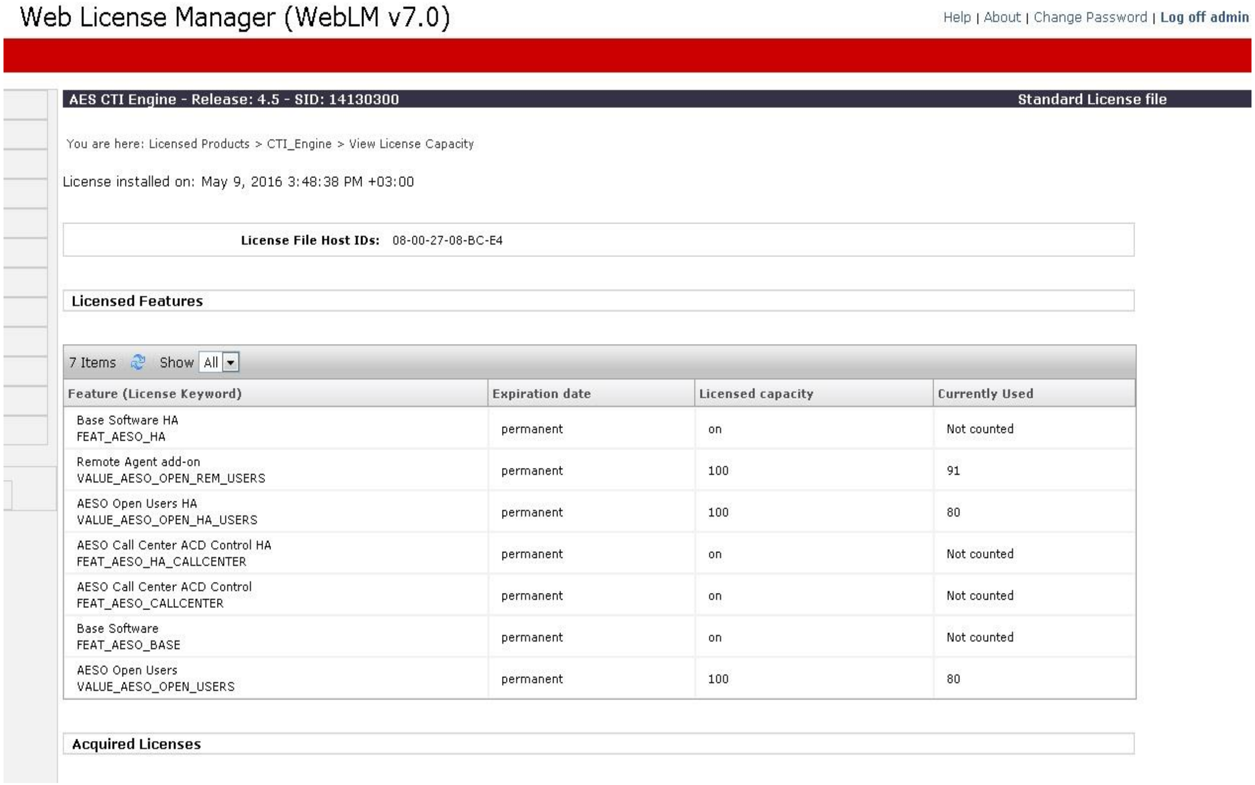
Avaya Global License Terms & Conditions

AVAYA GLOBAL SOFTWARE LICENSE TERMS
REVISED: March 2015

THIS END USER LICENSE AGREEMENT ("SOFTWARE LICENSE TERMS") GOVERNS THE USE OF PROPRIETARY SOFTWARE AND THIRD-PARTY PROPRIETARY SOFTWARE LIC THROUGH AVAYA. READ THESE SOFTWARE LICENSE TERMS CAREFULLY, IN THEIR ENTIRETY, BEFORE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (AS DEFINED IN SECTION A BELOW). BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF THE ENTITY FOR WHOM YOU ARE DOING SO (HEREINAFTER REFERRED TO INTERCHANGEABLY AS "YOU," "YOUR," AND "END USER"), AGREE TO THESE SOFTWARE LICENSE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT

○ Accept the License Terms & Conditions

Below is an example of what the licensing looks like once installed into WebLM.



Web License Manager (WebLM v7.0)

AES CTI Engine - Release: 4.5 - SID: 14130300 Standard License file

You are here: Licensed Products > CTI_Engine > View License Capacity

License installed on: May 9, 2016 3:48:38 PM +03:00

License File Host IDs: 08-00-27-08-BC-E4

Licensed Features

7 Items Show All

Feature (License Keyword)	Expiration date	Licensed capacity	Currently Used
Base Software HA FEAT_AESO_HA	permanent	on	Not counted
Remote Agent add-on VALUE_AESO_OPEN_REM_USERS	permanent	100	91
AESO Open Users HA VALUE_AESO_OPEN_HA_USERS	permanent	100	80
AESO Call Center ACD Control HA FEAT_AESO_HA_CALLCENTER	permanent	on	Not counted
AESO Call Center ACD Control FEAT_AESO_CALLCENTER	permanent	on	Not counted
Base Software FEAT_AESO_BASE	permanent	on	Not counted
AESO Open Users VALUE_AESO_OPEN_USERS	permanent	100	80

Acquired Licenses

The above shows that there are 100 standard users configured (AESO Open Users) and 100 high availability users (AESO Open Users HA). In addition we can see that call center functionality has been enabled (AESO Call Center ACO Control).

The CTI Engine will operate in one of three modes: Licensed mode, Grace Period mode, or Unlicensed mode.

Licensed mode is when the CTI Engine has been properly configured to use WebLM, licenses have been successfully loaded into the WebLM, and the CTI Engine is able to acquire and use those licenses. This should be the normal mode of operation for the CTI Engine.

Grace Period mode is entered when the CTI Engine has successfully operated in Licensed mode before, but the WebLM server is no longer accessible, or the WebLM server no longer has the licenses loaded. The CTI Engine will operate in Grace Period for up to 30 days since the last day that licenses were successfully acquired from WebLM. If the period of time is exceeded before licenses can be successfully acquired from a WebLM server, the CTI Engine will transition to Unlicensed mode.

Unlicensed mode is used when the above Grace Period has been exceeded, or if licenses have never been successfully acquired from a WebLM server. In this mode, the CTI Engine will have full functionality, but only for a total of five users. Unlicensed mode does not expire.

Connection to the WebLM Server is configured in the **`ctiengineconfig.xml`** file, as detailed in the CTI Engine Configuration section.

Certificate Configuration

The agent application uses encrypted web sockets (wss) to communicate with the AES CTI Engine. To support the use of wss, you must install a server certificate. You can install any of the following types of certificates in the server:

- Self-signed certificate
- Third-party or signed certificate from a request file
- Signed certificate with a private key

The mentioned certificates apply to all of the applications that are part of the AES CTI Engine solution including the CTI Engine and the CTI Engine Administration program.

Self-Signed Certificate

A self-signed certificate is created and signed by the server hosting the AES CTI Engine. Self-signed certificates do not contain a valid signature. As a result, whenever the client uses the AES CTI Engine, a security warning is displayed. Alternatively, to avoid the security warning, the certificate can be imported by each user’s certificate store. While acceptable during initial testing, this is not recommended for long term or production use. It is expected that any self-signed certificate used will be replaced with a signed certificate.

Creating a self-signed certificate and the associated files does not require root privileges. The self-signed certificate can be created using the default user. While creating a self-signed certificate, a password is asked. It is recommended to use a single password throughout the process.

Procedure

1. Create a new directory.
   For example, `/opt/avaya/certs`
2. Run the following command to navigate to the newly created directory:

   ```
   cd /opt/avaya/certs
   ```

3. Run the following command to generate the `private_key.pem` file:

   ```
   openssl genrsa -des3 -aes256  -out private_key.pem 2048
   ```

4. Type a password and type the same password as a confirmation password to create the file.
5. Run the following command to create the certificate request:

   ```
   openssl req -new -sha256 -key private_key.pem -out
   certificate_req.csr
   ```

6. Type the password created in step 4 on request.

   The system then requests to provide details for the following questions:

   a. **Country Name**: Type the two digit code for your country (e.g. “US” for the United States).

   b. **State or Province Name**: Type the name of your state or province (e.g. “California”).

   c. **Locality Name**: Type the name of your city or municipality (e.g. “Santa Clara”).

   d. **Organization Name**: Type the name of your company (e.g. “Avaya, Inc.”).

   e. **Organizational Unit Name**: Type the name of your organization (e.g. “IT”).

   f. **Common Name**: Type the fully qualified domain name (FQDN) of this server. (see below notes)

   g. **Email Address**: Press the Enter key for no value.

   h. **Challenge Password**: Press the Enter key for no value.

   i. **Optional Company Name**: Press the Enter key for no value.

   The system generates the `certificate_req.csr` file.

Note

If you are configuring multiple machines in the same computer domain, for example `cti1.company.com` and `cti2.company.com`, then you may use a wildcard certificate. To do so, replace the server name with the asterisk (*) character, leaving the domain portion of the name. For example, given the two example names above, a wildcard name of `*.company.com`, would apply to both servers. This will allow multiple servers to use the certificate and only require users to import a single certificate.

The CN (common name) is very important. It must match across the server, the corporate DNS, and the URL used to access the CTI Engine. The server must recognize the name as its own. The name must be registered in the Corporate DNS, so it can be resolved by all of the user's workstations. The URL configured in the call center definition must use this name. All of these names must match the CN configured in the certificate. (Keeping in mind the possibility of wildcard certificates, as noted above.)

7. Run the following command to remove the password from the private key:

   ```
   mv private_key.pem private_key.pem.org
   ```

   ```
   openssl rsa -in private_key.pem.org -out private_key.pem
   ```

   The reason for removing the password from the private key is to facilitate the interface from the web application in the user's browser into the CTI Engine. By removing the password, it is not prompted each time the web socket connection is made, and the connection will work seamlessly.

8. Run the following command to generate a self-signed certificate valid for a year:

   ```
   openssl x509 -req -sha256 -days 365 -in certificate_req.csr -
   signkey private_key.pem -out certificate.crt
   ```

   To change the amount of time the certificate will be valid, adjust the value for `days`.

9. Copy the files **`certificate.crt`** and **`private_key.pem`** to the **`/opt/avaya/ctiengine/certificates`** directory.

Now you can use secure web sockets with the CTI Engine.

You must copy a self-signed certificate into each user's workstation. The self-signed certificate should be imported into the Windows Certificate Store for Internet Explorer and Google Chrome or imported as an exception directly into Firefox. This allows the browser to treat the self-signed certificate as a genuine certificate from a trusted entity.

For the self-signed certificate to be properly recognized when imported into the Windows Certificate Store, the certificate must be installed in the Trusted Root Certification Authorities folder. Placing the self-signed certificate in any other store will not allow it to be recognized.

These actions are not required for a signed certificate.

Signed Certificate

Creating a signed certificate and the associated files does not require root privileges. A signed certificate can be created using the default user. While creating a signed certificate, a password is asked. It is recommended to use a single password throughout the process.

This process assumes a private key and certificate request file for the signed certificate are being created. If a self-signed certificate has already been created, the private key and certificate request files already generated can be reused if desired.

Procedure

1. Create a new directory.
   For example, **`/opt/avaya/certs`**

2. Run the following command to navigate to the newly created directory:

   ```
   cd /opt/avaya/certs
   ```

3. Run the following command to generate the **`private_key.pem`** file:

   ```
   openssl genrsa -des3 -aes256 -out private_key.pem 2048
   ```

4. Type a password and type the same password as a confirmation password to create the file.

5. Run the following command to create the certificate request:

   ```
   openssl req -new -sha256 -key private_key.pem -out
   certificate_req.csr
   ```

6. Type the password created in step 4 on request.

   The system then requests to provide details for the following questions:

   a. **Country Name**: Type the two digit code for your country (e.g. “US” for the United States).

   b. **State or Province Name**: Type the name of your state or province (e.g. “California”).

   c. **Locality Name**: Type the name of your city or municipality (e.g. “Santa Clara”).

   d. **Organization Name**: Type the name of your company (e.g. “Avaya, Inc.”).

   e. **Organizational Unit Name**: Type the name of your organization (e.g. “IT”).

   f. **Common Name**: Type the fully qualified domain name (FQDN) of this server. (see below notes)

   g. **Email Address**: Press the Enter key for no value.

   h. **Challenge Password**: Press the Enter key for no value.

   i. **Optional Company Name**: Press the Enter key for no value.

   The system generates the **`certificate_req.csr`** file.

Note

If you are configuring multiple machines in the same computer domain, for example **`cti1.company.com`** and **`cti2.company.com`**, then you may use a wildcard certificate. To do so, replace the server name with the asterisk (*) character, leaving the domain portion of the name. For example, given the two example names above, a wildcard name of **`*.company.com`**, would apply to both servers. This will allow multiple servers to use the certificate and only require users to import a single certificate.

The CN (common name) is very important. It must match across the server, the corporate DNS, and the URL used to access the CTI Engine. The server must recognize the name as its own. The name must be registered in the Corporate DNS, so it can be resolved by all of the user’s workstations. The URL configured in the call center definition must use this name. All of these names must match the CN configured in the certificate. (Keeping in mind the possibility of wildcard certificates, as noted above.)

7. Send the **certificate_crt.csr** file to the IT security department so that the signed certificate can be issued. Depending on the processes involved, this could take multiple days.
8. Once the certificate is received, place the file into the **/opt/avaya/certs** directory on the server. Make a copy of the file named **certificate.crt.**

   Note

   Use the Base64 encoding when creating the signed certificate. If the DER encoding is used, the certificate will not match up with the private key. To switch the encoding, use the following command:

   ```
   openssl x509 -inform der -in signed_cert.der -out certificate.crt
   ```

   Also, if the signed certificate is a compound certificate (meaning it includes the server certificate, plus one or more intermediate certificates), it is very important to make sure the certificates are ordered correctly. The first certificate must be the server certificate, followed by the intermediate certificates in order. If it is not, the CTI Engine will not be work properly.
9. Run the following command to remove the password from the private key and avoid the repetitive password prompt:

   ```
   mv private_key.pem private_key.pem.org
   ```

   ```
   openssl rsa -in private_key.pem.org -out private_key.pem
   ```

   Note

   To change the amount of time the certificate will be valid, adjust the value for `days`.
10. Copy the files **certificate.crt** and **private_key.pem** to the **/opt/avaya/ctiengine/certificates** directory.

Now you can use secure web sockets with the CTI Engine.

Signed Certificate with Key

In case where IT security teams do not allow you to create keys or certificate request files, then you must send the Server FQDNs to the security team to perform these actions. The IT security team will provide both private key and a signed certificate in the form of a single key store file. When sent this way, the key store file must be sent in the PKCS12 format. Such files will usually have the “pfx” extension.

Note that a pass phrase is usually associated with the private key file. This pass phrase must be provided with the files in order to access the certificate and private key.

Procedure

1. Create a new directory.
   For example, **/opt/avaya/certs**
2. Run the following command to navigate to the newly created directory:

```
   cd /opt/avaya/certs
   ```

3. Copy the key store file provided into **/opt/avaya/certs**. The following examples assume that the key store file is named **certificate.pfx**.
4. Extract the private key file with the following command:
   ```
   openssl pkcs12 –in certificate.pfx –out private_key.pem –nodes –
   nocerts
   ```
5. Extract the certificate file with the following command:
   ```
   openssl pkcs12 –in certificate.pfx –out certificate.crt –nodes -
   nokeys
   ```

   Note

   Also, if the signed certificate is a compound certificate (meaning it includes the server certificate, plus one or more intermediate certificates), it is very important to make sure the certificates are ordered correctly. The first certificate must be the server certificate, followed by the intermediate certificates in order. If it is not, the CTI Engine will not work properly.
6. If the private key has a password, run the following command to remove the password from the private key and avoid the repetitive password prompt:
   ```
   mv private_key.pem private_key.pem.org
   ```
   ```
   openssl rsa -in private_key.pem.org -out private_key.pem
   ```
7. Copy the files **certificate.crt** and **private_key.pem** to the **/opt/avaya/ctiengine/certificates** directory.

Now you can use secure web sockets with the CTI Engine.

Chapter 4: Configuring the CTI Engine

You may configure the CTI Engine and the Client Logger application using the Administrative web application as well as through the command line. Note that, you must perform all activities in this section using the `avaya` user.

CTI Engine Prerequisites

- Copy the **`ctienginelicense`** file in the **`/opt/avaya/ctiengine`** directory.
- Copy the **`license_public_key.pem`** file in the **`/opt/avaya/ctiengine/certificates`** directory.
- Copy the **`certificate.crt`** and **`private_key.pem`** files into the **`/opt/avaya/ctiengine/certificates`** directory.

Note that all of the above file names are required. If any of them are named something else, the files will not be found by the CTI Engine, and the CTI engine will not work.

CTI Engine Configuration

The configuration details for the CTI Engine are stored in the file **`ctiengineconfig.xml`**. This file is located at **`/opt/avaya/ctiengine`**. This file is a single line of xml and has the following key tags.

- **PORT**: Use the default port that is 8090, unless it requires change.
- **CONNECTIONTIMEOUT**: The number of minutes after which an inactive connection will be terminated. The only activity that is considered for this purpose is from the client to the CTI Engine. If the session is idle for longer than that, the CTI Engine will terminate the session, and the user will have to log back into the softphone to make it work. If users are away from the desks for longer periods of time while wishing to remain logged in, this value will need to be increased. The default value is 60 minutes. Note that this is not the only timer involved. Other timers could also affect the operation during the idle time, like any client-side timeout or any other network timers that are involved.
- **LOGOUTONTIMEOUT**: This determines if the agent is logged out of the Communication Manager ACD if a connection is timed out. If set to “n”, the user’s ACD login is completely unaffected and nothing happens when the connection is terminated. If set to “y”, the CTI Engine will attempt to log the user’s Agent ID is out of the ACD when the connection is terminated. It is possible for this attempt to fail, depending on the state of the user’s station at that time. This setting has no effect on users that are not logged into the ACD.
- **LOADBALANCER**: This is the IP address or list of IP addresses of any load balancers working with the CTI Engine. The CTI Engine rejects multiple connections for a single source, and this tag provides an exception in the case of load balancers. Enter the IP address of the load balancer into the parameter and the CTI Engine will allow for

multiple connections from that IP address. If multiple IP addresses are entered, they must be listed in this single parameter, separated by a comma.

- **TSAPIPRIMARYLINK**: The primary TSAPI TLink. The primary TLink will always be used if it is present.
- **TSAPISECONDARYLINK**: The secondary TSAPI TLink.
- **TSAPIUSERNAME**: The username of the CT User used to authenticate with the AES.
- **TSAPIPASSWORD**: The encrypted password of the CT User used to authenticate with the AES. To generate the encrypted version of the password, the utility **`aes-encryption`** is provided in the **`/opt/avaya/ctiengine`** directory. Run the utility and provide the password when prompted. The utility's output is the encrypted string for the password. Store this output string as the value for **TSAPIPASSWORD.**
- **ANIMASKING**: The type of the ANI masking encoding to use. The **ANIMASKING** value determines how the ANI masking value is to be encoded. The options are:
  - **A**: This indicates that it is to be encoded using the old method. This old method is not compatible with SIP trunks, but will work with any version of the Communication Manager. It is the only option available for Communication Manager versions older than 6.
  - **B**: This indicates that it is to be encoded using the newer method compatible with SIP trunks, but uses the broken encoding that was initially implemented. This is needed for most versions of the Communication Manager 6.x.
  - **C**: This indicates that it is to be encoded using the proper version of the newer method. This version works with SIP trunks, and is needed for the very last versions of Communication Manger version 6.x, and for Communication Manager version 7.x.
- **LOGSEVERITY**: The level of logging required. The options are:
  - Low: Use this option for the normal operation.
  - High: Use this option when you want to diagnose issues or problems.
- **LOGSIZE**: The maximum size of a given log file. If the size is exceeded, the log file will roll over to a new file.
- **LOGRETENTION**: The maximum number of days for which logs are kept.
- **WEBLMURL**: Contains the URL used to access the WebLM server. The default format should be `https://x.x.x.x:52233/WebLM/LicenseServer`. Before starting the CTI Engine with this setting in place, it is recommended to test the URL from a browser to ensure that the URL is correct. When accessed from a browser, the browser will be directed to a login page for WebLM.
- **PRIMARYCTI**: Specifies if this system will act as a primary server, or if it will act as a stand-by secondary server when the primary server is unavailable. It should be set to "yes" when this system is the primary server, and set to "no" when this system is the stand-by secondary server.
- **USERS**: This is the number of concurrent users this server can access if PRIMARYCTI is set to "yes". This cannot be greater than the number of users specified in WebLM. If

multiple CTI Engines will share the load, the total number of users defined across all CTI Engines cannot be greater than the number of users specified in WebLM. If PRIMARYCTI is set to “no”, this is set to “0”.

- **HA_USERS**: This is the number of concurrent stand-by users this server can access if PRIMARYCTI is set to “no”. This cannot be greater than the number of stand-by users specified in WebLM. If multiple CTI Engines will share the load, the total number of stand-by users defined across all CTI Engines cannot be greater than the number of stand-by users specified in WebLM. If PRIMARYCTI is set to “yes”, this is set to “0”.

These parameters directly correspond to the entries in the CTI Engine screen of the administration program.

Managing the CTI Engine

The program **`ctiengineservice`** is used to control the CTI Engine. The primary commands to run from the **`/opt/avaya/ctiengine`** directory location are:

- **`./ctiengineservice start:`** Use this command to start the CTI Engine.
- **`./ctiengineservice stop:`** Use this command to stop the CTI Engine.
- **`./ctiengineservice status`**: Use this command to check whether the CTI Engine is currently running.

Client Logger Configuration

The configuration details for the Client Logger are stored in the file **`clientloggerconfig.xml`**. This file is located at **`/opt/avaya/ctiengine`**. This file is a single line of xml and has the following key tags.

- **PORT**: Use the default port that is 9090, unless it requires.
- **CONNECTIONTIMEOUT** The number of minutes after which an inactive connection will be terminated.
- **LOADBALANCER**: This is the IP address or list of IP addresses for load balancers. The CTI Engine rejects multiple connections for a single source, and this tag provides an exception in the case of load balancers.
- **LOGSIZE**: The maximum size of a given server log file. If the size is exceeded, the log file will roll over to a new file.
- **LOGRETENTION**: The maximum number of days for which server logs are retained.
- **USERLOGSIZE**: The maximum size of a user log file. If the size is exceeded, the log file will roll over to a new file.
- **USERLOGRETENTION**: The maximum number of days for which user log files are retained.

These parameters directly correspond to the entries in the **Client Logger** screen of the administration program. The detailed information provided for the **CONNECTIONTIMEOUT** and **LOADBALANCER** parameters for the CTI Engine also apply to those parameters for the Client Logger.

Managing the Client Logger

The program **`clientloggerservice`** is used to control the Client Logger. The primary commands to run from the **`/opt/avaya/ctiengine`** directory location are:

- **`./clientloggerservice start:`** Use this command to start the Client Logger.
- **`./clientloggerservice stop:`** Use this command to stop the Client Logger.
- **`./clientloggerservice status`**: Use this command to check whether the Client Logger is currently running.

Chapter 5: Upgrading the CTI Engine Application

Before beginning

1. Using the user `avaya`, create the directory **`/opt/avaya/conf`** if it does not already exist.
2. Copy the following files to the folder **`/opt/avaya/conf`**:
   - **`/opt/avaya/ctiengine/ctiengineconfig.xml`**
   - **`/opt/avaya/ctiengine/clientloggerconfig.xml`**

   NOTE

   These actions ensure that the configuration files needed by the CTI Engine are saved in a common location so that they are not lost after performing the upgrade.
3. Some files are only in newer updates, for example **`aesloggerconfig.xml`** or **`clientloggerconfig.xml`**. If they are not present in the older version, skip the file and continue.
4. The name of the directory used by the CTI Engine has changed. The details are as follows:

Previous name	**Current name**
`/opt/avaya/aessfdc`	**`/opt/avaya/ctiengine`**
`websocketserver`	**`ctiengineservice`**
`websocketctiserverconfig.xml`	**`ctiengineconfig.xml`**
`aesloggerservice`	**`clientloggerservice`**
`aesloggerconfig.xml`	**`clientloggerconfig.xml`**

Procedure

1. Use the `avaya` username credentials.
2. Change the directory to **`/opt/avaya/ctiengine`** and run the following commands:

```
./ctiengineservice stop
./clientloggerservice stop
```

This will stop the CTI Engine and Client Logger programs. It is possible that the version to be removed does not have the Client Logger program. In such case, ignore the respective command.

3. Run the following command with administrative privileges:

   ```
   sudo rpm -e AES-Connector-CTIEngine-4.0-10.i686
   ```

   The system uninstalls the old CTI Engine.

   ✪ **Note**

   The package could be a different version. If the old version is not known, use the command **`rpm -qa | grep AES`** to find the old version.

4. Run the following command to install the new CTI Engine with administrative privileges:

   ```
   sudo rpm -ivh AES-Connector-CTIEngine-4.5.1-<rev>.i686.rpm
   ```

5. Complete the rest of the steps using the `avaya` credentials.
6. Copy the file **`/opt/avaya/conf/ctiengineconfig.xml`** to **`/opt/avaya/ctiengine`**.

   ✪ **Note**

   Ensure that the options have not changed, or there are no new parameters added between the versions before overwriting the file. If they have, you must manually update the respective file.

7. Copy the file **`/opt/avaya/conf/clientloggerconfig.xml`** to **`/opt/avaya/ctiengine.`**

   ✪ **Note**

   Ensure that the options have not changed, or there are no new parameters added between the versions before overwriting the file. If they have, you must manually update the respective file.

8. Copy the `file` **`/opt/avaya/conf/ctienginelicense`** `to` **`/opt/avaya/ctiengine.`**

   ✪ **Note**

   If the CTI Engine package being replaced is **`AES-SF-Connector-CTIEngine-2.5-1`** or older, the old **`ctienginelicense`** file cannot be used and a new license file must be obtained. If the CTI Engine package being replaced is **`AES-Connector-CTIEngine -2.6-1`** or newer, the **`ctienginelicense`** file will work with the updated version.

9. Copy the file **`/opt/avaya/conf/license_public_key.pem`** to **`/opt/avaya/ctiengine/certificates`**.
10. Copy the files **`/opt/avaya/certs/certificate.crt`** and **`/opt/avaya/certs/private_key.pem to /opt/avaya/ctiengine/certificates`**.
11. Change the directory to **`/opt/avaya/ctiengine`**.
12. Run the following command to start the CTI Engine:

```
./ctiengineservice start
```

13. Run the following command to start the logger service:

    ```
    ./clientloggerservice start
    ```

Appendix A: Support Files

If issues with the CTI Engine are found, then when reporting these issues to Avaya Application Support, please always supply the log files that cover the time period of the incidence. The log files are stored at the following location:

`/opt/avaya/ctiengine/log`

The CTI Engine creates rolling logs that capture all of its activity. The base name of the file is ctiengine_<date>_<ordinal>.txt. The <date> element is in the form of day of the month, month, and year. The <ordinal> number is usually 0, but will increment if multiple files for a single day are created. For example, the first log file on 24 April 2015 would be:

`ctiengine_24042015_0.txt`

The Client Logger has rolling log files that capture its own status and administrative activity. It also accepts logging from the users and stores them for each user, capturing the same information as the browser console. The base log file uses the same naming as the CTI Engine logs, but in the form of clientlogger_<date>_<ordinal>.txt, where <date> and <ordinal> use the form described above. For example, the second log file on 24 April 2015 would be:

`clientlogger_24042015_1.txt`

The user log files stored by the Client Logger are all stored in directories that are named in the form <date> for the date the log takes place. The files within those directories are named after the extension the log file represents, in the form `<extention>_<ordinal>.txt`. For example, all user log files for the date of 24 April 2015 will be stored in a folder named **`24042015`**. Within that directory, the first file for the extension 56911 would be:

`56911_0.txt`

For any log files from the CTI Engine, simply gather the log files together and send them to Avaya Application Support.

Appendix B: Failover

There are multiple levels of failover in AES CTI Engine that the CTI Engine uses to communicate with the CM.

There are two ways to handle AES failover in the CTI Engine. The first way is to simply configure an HA AES TLink into the CTI Engine configuration file. Even though only a single TLink is configured, it represents an HA pair, so the desired redundancy is handled automatically by the HA AES pair, rather than by the CTI Engine. The second way is to configure two TLinks into the CTI Engine configuration file. In this case, the CTI Engine will directly manage the failover. The CTI Engine will always use the primary TLink when it is available, and only use the secondary TLink if the primary is unavailable. However, it will not switch between the two unless the current connection fails. So, once it switches to the second TLink, it will continue to use it until either the CTI Engine is restarted, or the TLink fails, forcing the CTI Engine to reconnect.

Also, it is entirely possible to use both mechanisms, by making the primary TLink an HA TLink, and the secondary TLink either a simple TLink or another HA TLink.

Using an extra CTI Engine server allows for failover at the CTI Engine level. If the client interface allows for the definition of multiple CTI Engine servers, then two servers can be used in parallel. In general, one will be designated the primary server and one the secondary server, where the secondary server is only used if the primary server is unavailable. The configuration for this is dependent on the client application.

An alternative method of providing redundancy for the Open CTI Server is to make use of an HTTP load balancer. When configuring an HTTP load balancer, there are a couple important considerations. The first is that session affinity must be enabled. Once a user makes a connection, the load balancer must ensure that the user's session is maintained over a single CTI Engine Server. The second is that the HTTP load balancer must also be able to handle web sockets. The connections from the client application to the CTI Engine and to the Client Logger both use web sockets, and that protocol must be supported.