Installation Manual

AudioCodes WebRTC Solutions for Enterprises

WebRTC Client Installation on HTTP Server

Version 2.6.0

Notice

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document can be downloaded from https://www.audiocodes.com/library/technical-documents.

This document is subject to change without notice.

Date Published: January-02-2024

Customer Support

Customer technical support and services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our website at https://www.audiocodes.com/services-support/maintenance-and-support.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our website at https://online.audiocodes.com/documentation-feedback.

Stay in the Loop with AudioCodes

Related Documentation

Document Name
https://www.audiocodes.com/solutions-products/solutions/enterprise-voice/webrtc-connectivity
WebRTC Softphone Client Quick Guide
WebRTC Softphone User's Manual
WebRTC Click-to-Call Widget Installation and Configuration Guide
WebRTC Android Client SDK API Reference Guide

Document Name
WebRTC iOS Client SDK API Reference Guide
WebRTC Web Browser Client SDK API Reference Guide

Document Revision Record

LTRT	Description
14120	Initial document release (Ver. 1.1).
14121	NGINX file settings and reload file commands.
14122	Typo (in footer).
14123	Update for Version 1.2; HTTP basic authentication for NGINX and Apache.
14124	Updated Sections: Solution Overview; Advanced Options Configuration File; Added Sections: Client Logo Customization; Deploying AudioCodes WDE Extension.
14125	Update for Version 1.5; Oauth configuration.
14126	IIS HTTP server configuration and installation.
14127	Update for Version 2.2; additional advancedOptions customization options, and deploying AudioCodes WDE Extension using an installation wizard.
14128	sipServerAddress configuration.
14129	Upgrade instructions for WebRTC.
14150	Updated for Version 2.4; configuration on auto-play, additional OAuth configurations and customer-info; browser auto-play policy configuration.
14151	Configuration on SDK patch modes.
14152	Configuration for VDI integration, media device selection, and ACD Client Configuration Not-Ready state values.
14153	Configure Citrix Remote Desktop for Integration with

LTRT	Description
	the WebRTC Client.
14154	Troubleshooting client connections; configuration for Agent Assist, codec filtering, and additions to media device selection; customizing sound files.
14155	Updated for Version 2.4.6; localVideofilter configuration; Virtual Background Images.
14156	Updated for Version 2.4.7; storageConfig and forceLoginPromptOnEmptyCache.
14157	Updated for Version 2.4.8. Updated Advanced Options Configuration File section, Advanced Options Configuration File section. Added Language and Text Customization section.
14158	Added the Upgrading Web Client to a New Release section. Updated Upgrading WebRTC Client on NGINX Server, Upgrading WebRTC Client on Apache Server and Upgrading WebRTC Client on IIS sections.
14159	Updated for Version 2.5.3. Added the *isAllowedToResumeLocalHoldByRemoteReInvite* attribute to Advanced Options Configuration File.
14160	Updated for Version 2.5.4. ■ It is no longer mandatory to deploy the client specifically at a directory named *webrtc_client*. ■ In the Configuring the HTTPS Site section, references to the deprecated Web Platform Installer were removed. ■ In the Advanced Options Configuration File section, the *disableEditing* property under *defaultServerConfig* was added. ■ In the Advanced Options Configuration File section, the *autoplayNotificationOptions* property has been deprecated and has been replaced with the *notificationOptions* property. ■ The Maintaining Web Client Activity When Browser Tab is in the Background section was added.

LTRT	Description
	■ The Troubleshooting Client Connections section was updated.
14161	Updated for Version 2.5.5. ■ Added support for Dual registration.
14162	Updated for Version 2.5.6. ■ Added support for No-answer Timeout configuration.
14163	Updated for Version 2.6.0. ■ Modified the **Customizing WebRTC Client** section: ✓ Deprecated and removed defaultOAuthConfig ✓ Deprecated and removed supportedAuthenticationSchemes ✓ Added the **Configure Authentication Providers for User Sign-In** sub-section, for supporting multiple authentication providers ✓ Added the **Configure Integrations with Contact Center Platforms** sub-section, for supporting multiple Contact Center integrations including Genesys PureCloud ✓ Added the **Migrating Authentication Providers and Contact Center Configuration from Previous Versions** section ■ Added the Configuring SBC WebRTC section

Table of Contents

1 Introduction

This document provides an overview of AudioCodes' WebRTC client solution and describes how to deploy and configure the WebRTC client on an HTTP-based server (NGINX or Apache).

2 Solution Overview

AudioCodes offers a WebRTC-based softphone client that can be used by agents in the Genesys PureEngage and PureConnect environment. This WebRTC softphone is used by agents, in conjunction with Genesys agent desktop applications - Workspace Desktop Edition (WDE), Workspace Web Edition (WWE) and Interaction desktop - as their telephony device that enables them to make and receive calls and perform other telephony functions.

This WebRTC-based phone is intended primarily for work-at-home agents and can also be used by agents located on the customer's premises.

As part of the solution, the Customer is required to embed a Web page within an existing Web server (it can be the Customer's main server, or a Web server dedicated to the WebRTC setup). This Web page will be used by agents to browse their Softphone page. Specific guidance on how to embed this page is described in Chapter 4.

In addition, the Customer is required to adjust the Genesys PureEngage DN object configuration, as described in this document.

The agent can refer to the WebRTC Web Softphone User's Manual for a description on using this WebRTC softphone.

Benefits

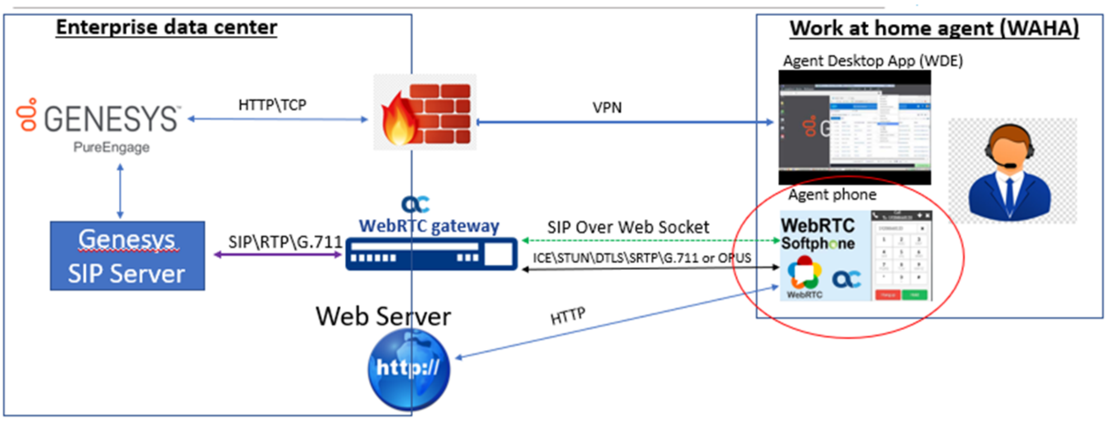
AudioCodes WebRTC client provides the following benefits:

- Ease of maintenance as there is no need to install and maintain softphone applications
- Enhanced voice quality using Opus codecs
- Removes requirement for VPN:
  - WebRTC uses ICE that can traverse NAT and firewalls
  - Secure and encrypted calls, using HTTPS, DTLS and SRTP
- Allows video calls (next phase)

Components

The following high-level diagram illustrates the components of this solution, which are also described in detail below.

Figure 2-1: Solution Components



- The following components are located at the Customer's data center:
    - Genesys PureEngage / PureConnect servers
    - WebRTC gateway used to convert WebRTC traffic from the agents to plain SIP and RTP traffic towards Genesys SIP server
    - Web server used for browsing to the page that displays the client softphone and downloads the WebRTC-based client JavaScript
- The following components are located on the agent's desktop:
    - Genesys agent desktop applications (WDE or WWE or Interaction desktop)
    - Google/Firefox-based client used for the WebRTC softphone application

High-level Call Flow

The call flow of this WebRTC client solution is as follows:

1. The agent browses to the Web server.
2. The browser downloads the client JavaScript that points to the SBC IP address.
3. The agent enters the agent's DN number, and associated credentials.
4. The client registers to the Genesys SIP server using the SIP protocol.
5. The SIP server authenticates the client using SIP digest authentication.
6. The SIP server pairs the client with the agent.
7. The agent can originate and receive calls, perform call transfers, conference calls, and hold and retrieve calls. In this first phase, all telephony actions control is performed from the agent's desktop application (third-party call control).

3 Installing WebRTC Client on HTTP Server

The WebRTC client needs to be installed on an HTTP-based server. This server can either be a dedicated HTTP server or one of your existing servers used for other applications (for example, your Web-hosting server).

This document describes the installation on the following popular HTTP servers:

- NGINX HTTP server (see Installing on NGINX HTTP Server below)
- Apache HTTP server (see Installing on Apache HTTP Server on page 7)
- Internet Information Services (IIS) for Windows (see Installing Internet Information Services (IIS) on Windows HTTP Server on page 9)

Installing on NGINX HTTP Server

The following describes how to deploy the WebRTC client on an NGINX HTTP server.

Prerequisites

Make sure you have fulfilled the following prerequisites:

- Operating system installed (tested with CentOS 7)
- NGINX installed
- DNS domain name for the server
- SSL certificates installed
- Redirect HTTP to HTTPS is configured

Configuring NGINX

The following describes the special configuration for NGINX. The configuration file is located in the /etc/nginx/nginx.conf directory.

> - TLSv1.2 and TLSv1.3 have been set as the supported protocols.
> - The cache needs to be controlled.
> - This configuration uses gzip compression.
> - This configuration supports IPv6 and the HTTP2 protocol.

➢ **To configure NGINX:**

1. Copy and paste the below configuration to the nginx.conf file. This contains all the basic settings required for AudioCodes WebRTC softphone client.

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

include /usr/share/nginx/modules/*.conf;

events {
   worker_connections 1024;
}

http {
   log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';

   access_log  /var/log/nginx/access.log  main;

   sendfile            on;
   tcp_nopush          on;
   tcp_nodelay         on;
   keepalive_timeout   65;
   types_hash_max_size 2048;

   include             /etc/nginx/mime.types;
   default_type        application/octet-stream;

   include /etc/nginx/conf.d/*.conf;


   server {
      listen       80;
      server_name    YOUR_SITE_NAME;
      return       301 https://$host$request_uri;
   }

   server {
      listen      443 ssl http2 default_server;
      listen      [::]:443 ssl http2 default_server;
      server_name    YOUR_SITE_NAME;

      ssl    on;
      ssl_certificate        /etc/pki/tls/certs/site_certificate.crt ;
      ssl_certificate_key    /etc/pki/tls/private/private.key;
```

```
        root         /var/html;

        gzip on;
        gzip_vary on;
        gzip_min_length 1024;
        gzip_proxied any;
        gzip_types text/plain text/css application/javascript application/json
text/xml application/xml text/javascript;

        add_header Cache-Control no-cache;
        expires 24h;
        ssl_session_timeout  10m;
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_prefer_server_ciphers on;

        include /etc/nginx/default.d/*.conf;

        location / {
          try_files $uri $uri/ =404;
        }

    }
}
```

2. Reload the NGINX server, using the following commands:

```
sudo nginx -t
sudo systemctl reload nginx
sudo systemctl status nginx
```

Deploying WebRTC Client on NGINX Server

The following describes how to deploy the WebRTC client on an NGINX HTTP server:

➢ **To deploy on NGINX server:**

1. Unzip the file webrtc-web-client-vx.x.x.zip, using the 7-Zip utility.
2. Copy the unzipped content to the */var/html/webrtc_client/* directory.

> ⚠ */var/html/* is not the default directory for NGINX.

3. Verify owners and permissions:

```
-rw-rw-r-- 1 centos  1819 Mar 23 17:46 asset-manifest.json
-rw-rw-r-- 1 centos 22382 Mar 23 17:46 favicon.ico
-rw-rw-r-- 1 centos  2529 Mar 23 17:46 index.html
-rw-rw-r-- 1 centos  8581 Mar 23 17:46 logo192.png
-rw-rw-r-- 1 centos centos 22920 Mar 23 17:46 logo512.png
-rw-rw-r-- 1 centos centos   722 Mar 23 17:46 logo-audiocodes.ico
-rw-rw-r-- 1 centos centos   492 Mar 23 17:46 manifest.json
-rw-rw-r-- 1 centos centos  1365 Mar 23 17:46 precache-
manifest.f4a95be0ec7e7fa979e757e1c258814f.js
-rw-rw-r-- 1 centos centos    57 Mar 23 17:46 robots.txt
-rw-rw-r-- 1 centos centos  1209 Mar 23 17:46 service-worker.js
drwxrwxr-x 5 centos centos    37 Mar 24 07:57 static
```

Restricting Access with HTTP Basic Authentication

To restrict access to the WebRTC client, you can configure basic authentication on the NGINX server for the client page, which obligates users to enter a preconfigured username and password when accessing the page.

➢ **To configure basic authentication on NGINX server:**

1. Follow the instructions in https://docs.nginx.com/nginx/admin-guide/security-controls/configuring-http-basic-authentication/.
2. Edit the NGINX configuration file, and set restricted access to the client installation path ("*/webrtc_client*" in our example):

```
location /webrtc_client {
        auth_basic "Restricted Content";
        auth_basic_user_file /etc/nginx/.htpasswd;
        try_files $uri $uri/ =404;
    }
```

Upgrading WebRTC Client on NGINX Server

➢ **To upgrade WebRTC Client on NGINX Server:**

To upgrade the WebRTC Client on the NGINX server, see Upgrading the Web Client to a New Release.

Installing on Apache HTTP Server

The following describes how to deploy the WebRTC client on an Apache server.

Prerequisites

Make sure you have fulfilled the following prerequisites:

- Operating system installed (validated with CentOS 7)
- Apache installed
- DNS domain name for the server
- SSL certificates installed
- Redirect HTTP to HTTPS is configured

Deploying WebRTC Client on Apache Server

The following describes how to deploy the WebRTC client on an Apache server:

➢ To deploy on Apache server:

1. Unzip the file webrtc-web-client-vx.x.x.zip, using the 7-Zip utility.
2. Copy the unzipped content to the */var/www/html/webrtc_client/ directory* (CentOS 7).
3. Verify owners and permissions:

```
-rw-rw-r-- 1 centos  1819 Mar 23 17:46 asset-manifest.json
-rw-rw-r-- 1 centos centos 22382 Mar 23 17:46 favicon.ico
-rw-rw-r-- 1 centos centos  2529 Mar 23 17:46 index.html
-rw-rw-r-- 1 centos centos  8581 Mar 23 17:46 logo192.png
-rw-rw-r-- 1 centos centos 22920 Mar 23 17:46 logo512.png
-rw-rw-r-- 1 centos centos   722 Mar 23 17:46 logo-audiocodes.ico
-rw-rw-r-- 1 centos centos   492 Mar 23 17:46 manifest.json
-rw-rw-r-- 1 centos centos  1365 Mar 23 17:46 precache-
manifest.f4a95be0ec7e7fa979e757e1c258814f.js
-rw-rw-r-- 1 centos centos    57 Mar 23 17:46 robots.txt
-rw-rw-r-- 1 centos centos  1209 Mar 23 17:46 service-worker.js
drwxrwxr-x 5 centos centos    37 Mar 24 07:57 static
```

HTTP Basic Authentication

To restrict access to the WebRTC client, you can configure basic authentication on the Apache server for the client page. This obligates users to enter a preconfigured username and password when accessing the page:

➢ To configure basic authentication for Apache server:

- *Edit /var/www/html/.htaccess*, and then set restricted access to the client installation path ("*/webrtc_client*" in our example):

```
SetEnvIf Request_URI ^/webrtc_client require_auth=true
Order Deny,Allow
Deny from all
Satisfy any
Require valid-user
Allow from env=!require_auth
AuthType Basic
AuthName "Restricted Content"
AuthUserFile /etc/httpd/.htpasswd
Require valid-user
```

Upgrading WebRTC Client on Apache Server

➢ **To Upgrade WebRTC Client on Apache Server:**

To upgrade the WebRTC Client on the Apache Server, see Upgrading the Web Client to a New Release.

Installing Internet Information Services (IIS) on Windows HTTP Server

The procedures below describe how to install the WebRTC client on an IIS server. All actions are performed from the Administrator account.

Prerequisites

Make sure you have fulfilled the following prerequisites:

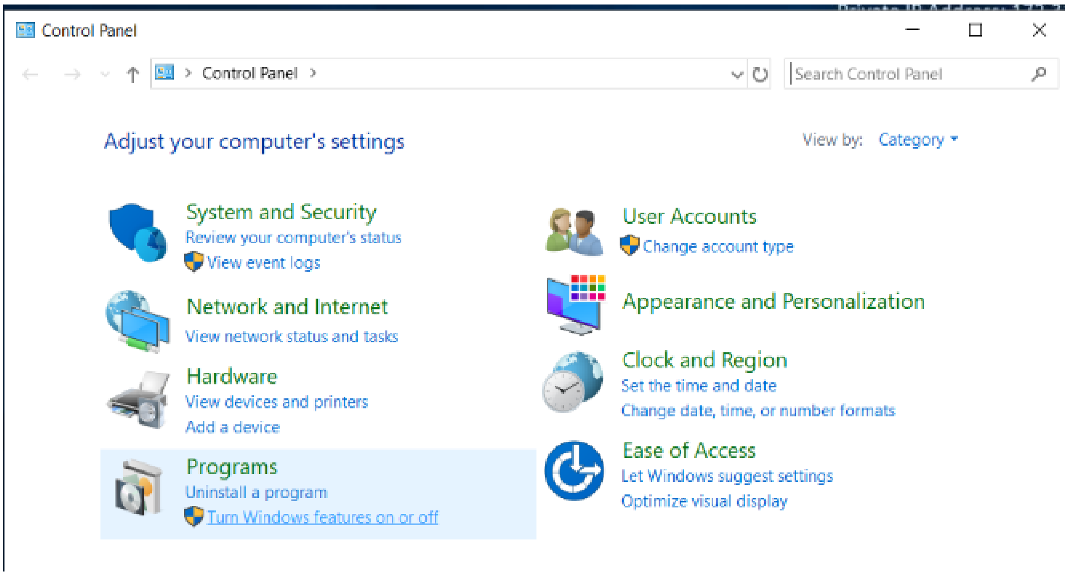
- Windows server has been installed (validated with Amazon Web Services cloud with Microsoft Windows Server 2019 Base)
- Cloud firewall (if the server is installed on the cloud) was configured for HTTP and HTTPS
- Public static IP address has been assigned to the Windows server
- Created a Fully Qualified Domain Name (FQDN) (DNS domain name) for the IP
- Configured a remote desktop connection for communication and file exchange with the server

Installing the IIS Server

The procedure below describes how to install the IIS server.

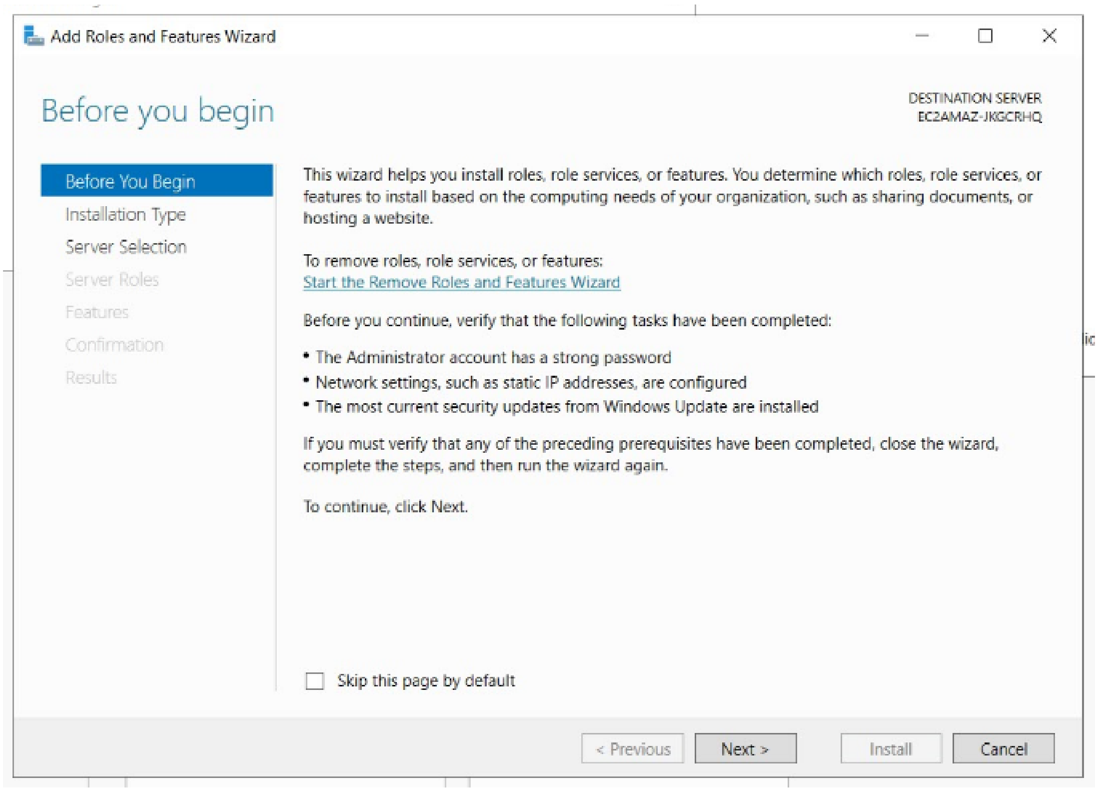
➢ **To install the IIS server:**

1. From the Windows Start menu, click **Control Panel**.

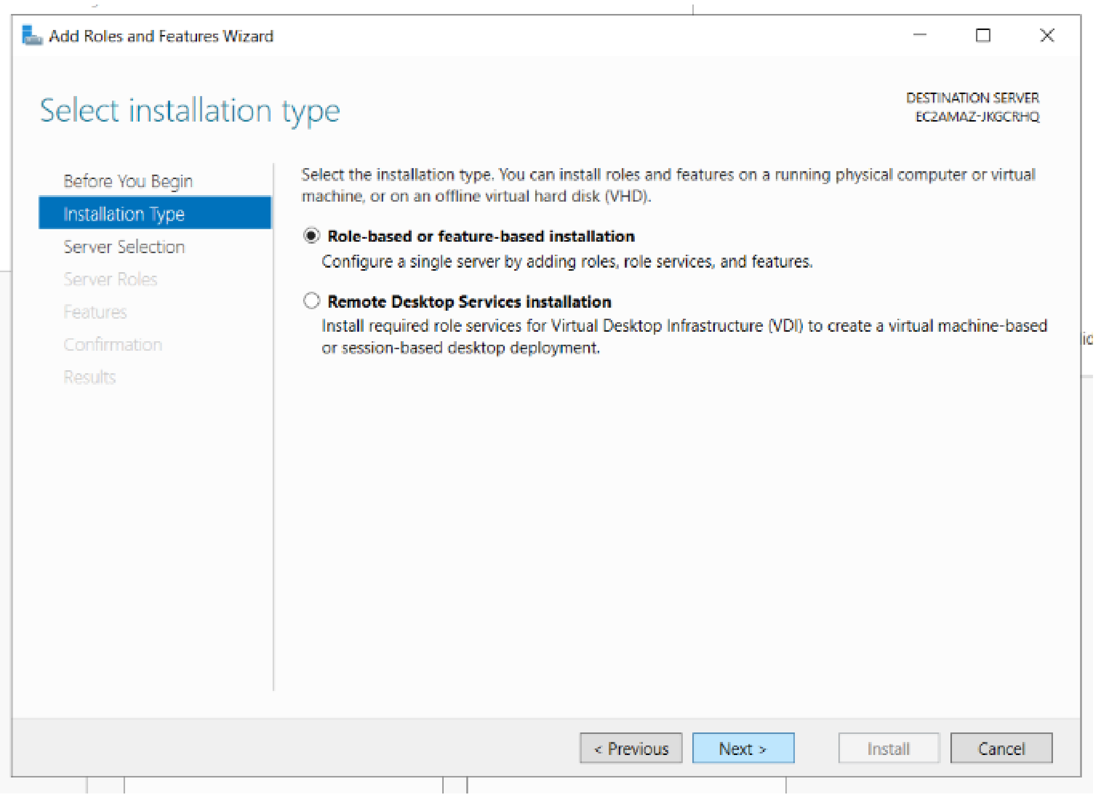


2. Click **Programs**, and then click **Turn Windows features on or off**.

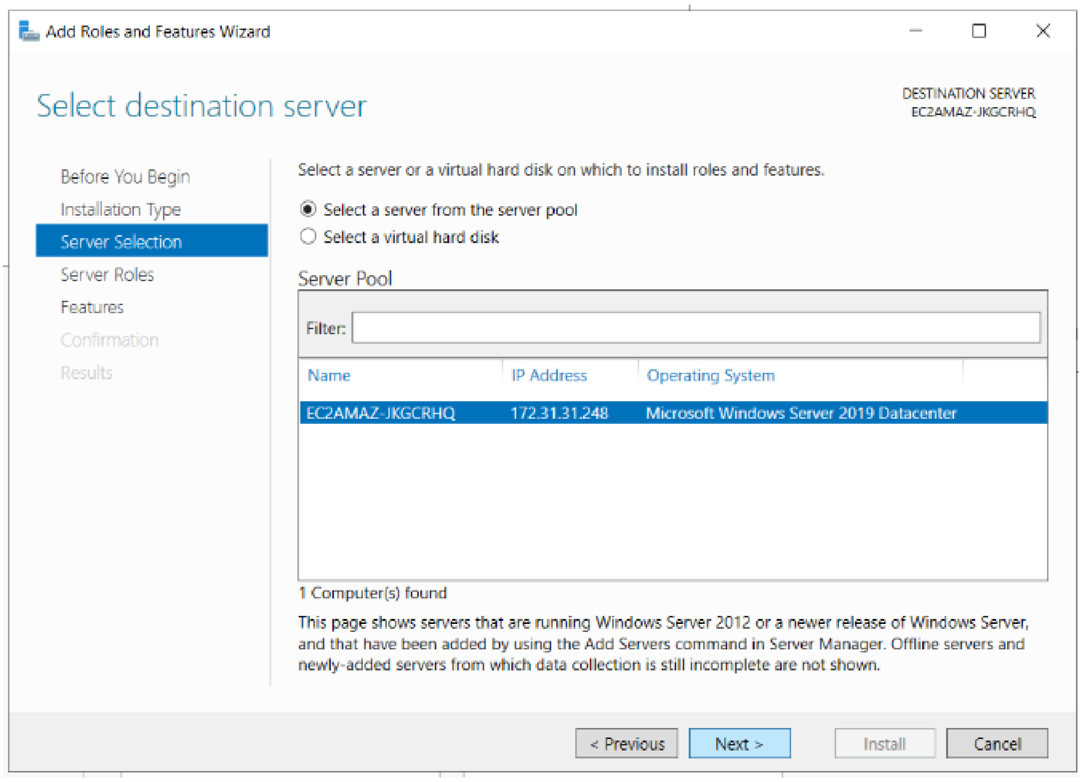
3. On the Before you Begin page, click **Next**; the Installation Type page appears.



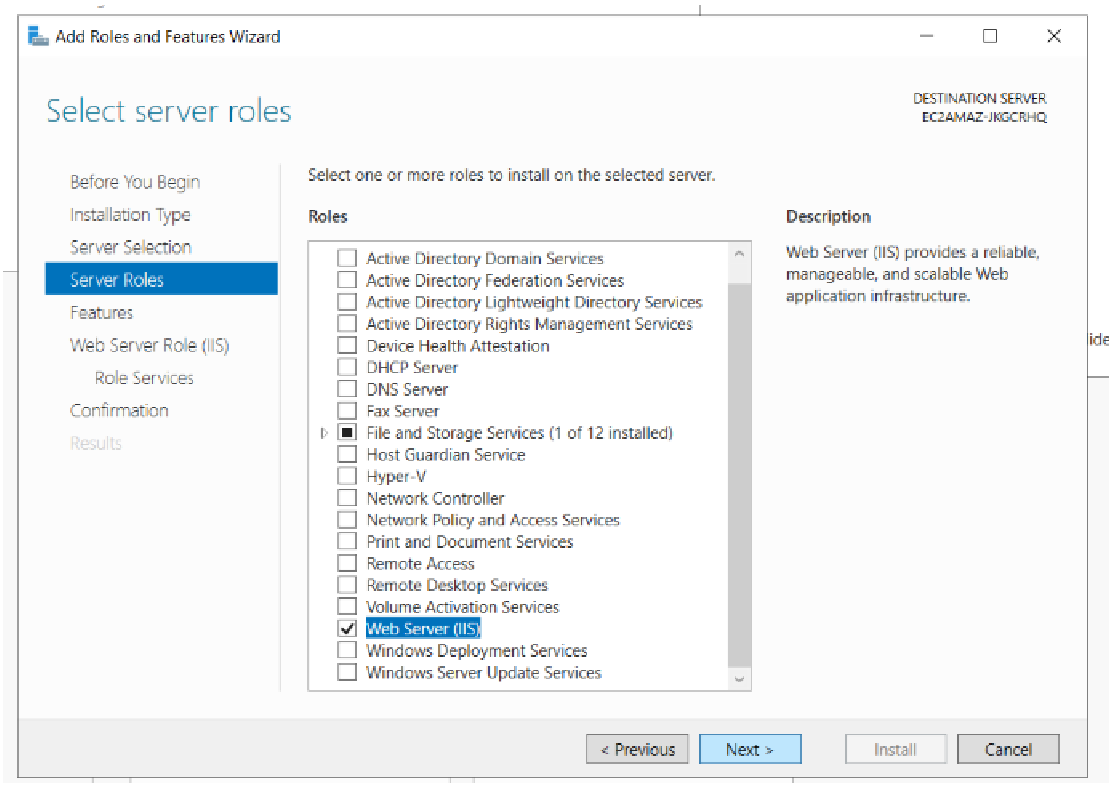
4. Select the Role based or feature-based installation option, and then click **Next**.



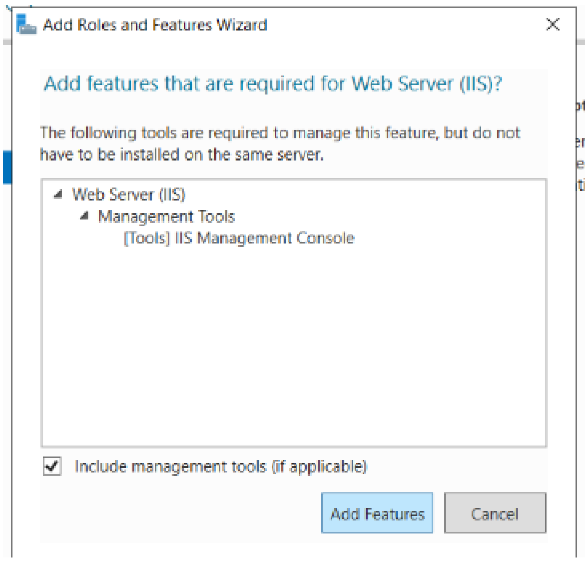
5. On the Select Destination Server page, select the server, and then click **Next**.



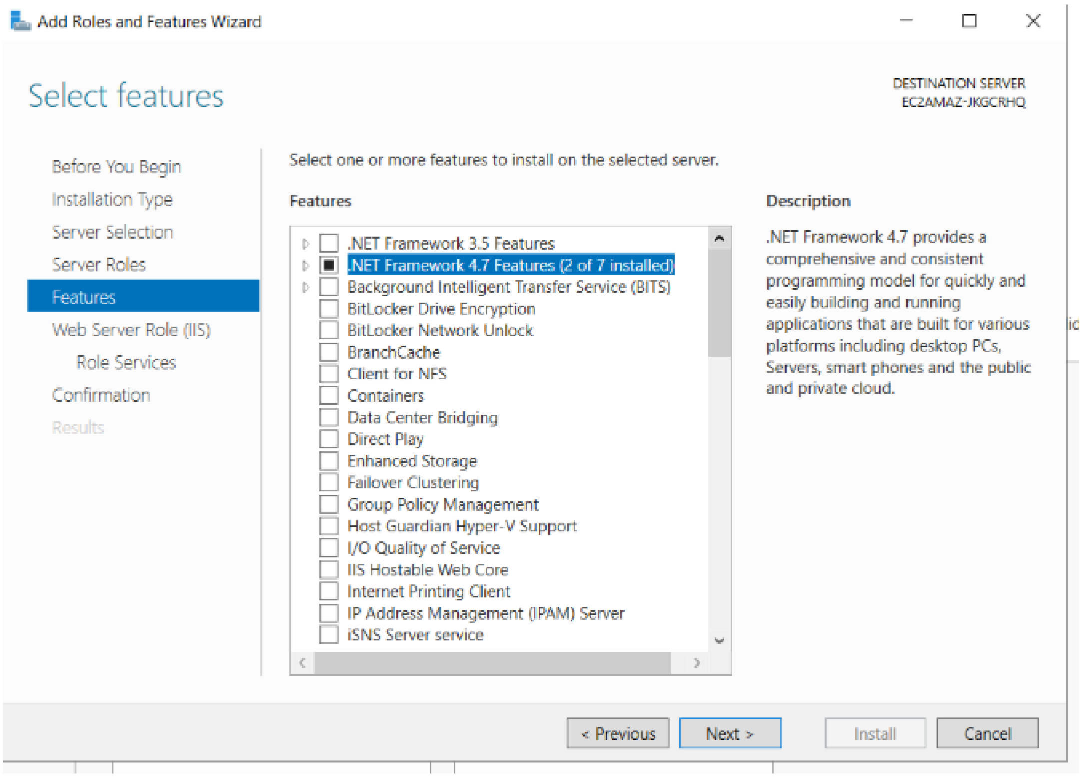
6. On the Select Server Roles page, select the 'Web Server IIS' check box.



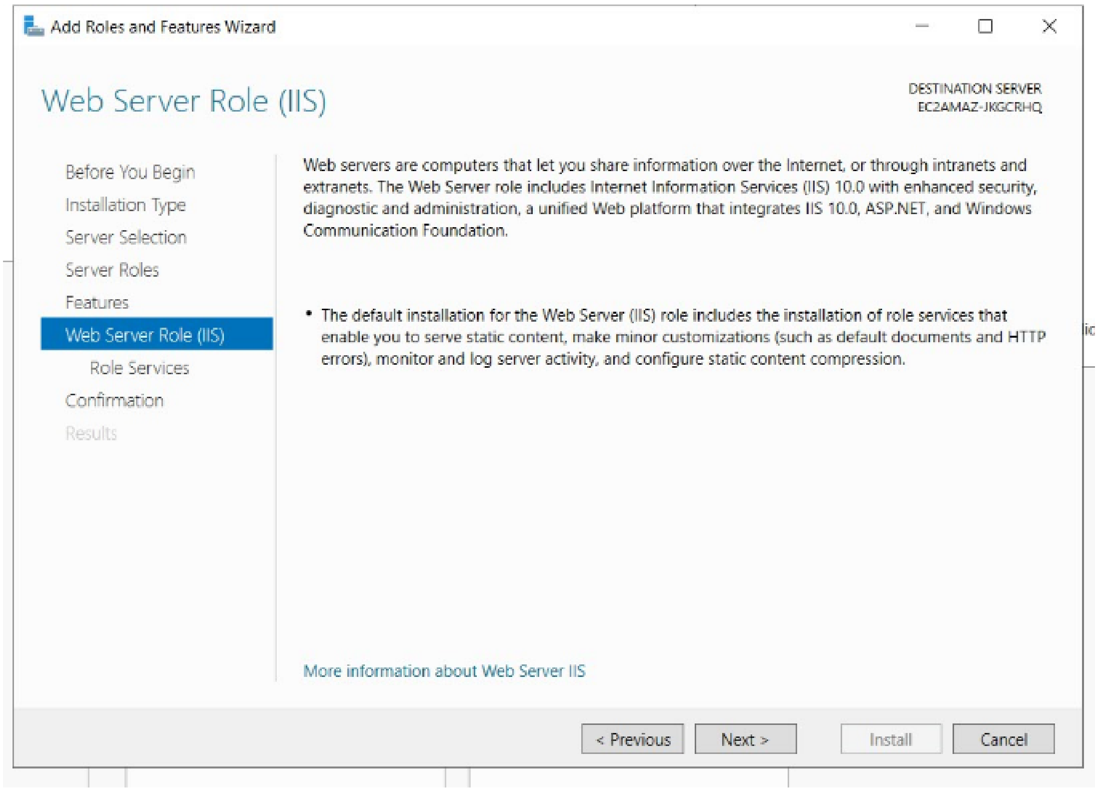
7. On the Add features that are required for Web server page, select the 'Include management tools' check box, and then click **Add Features**.



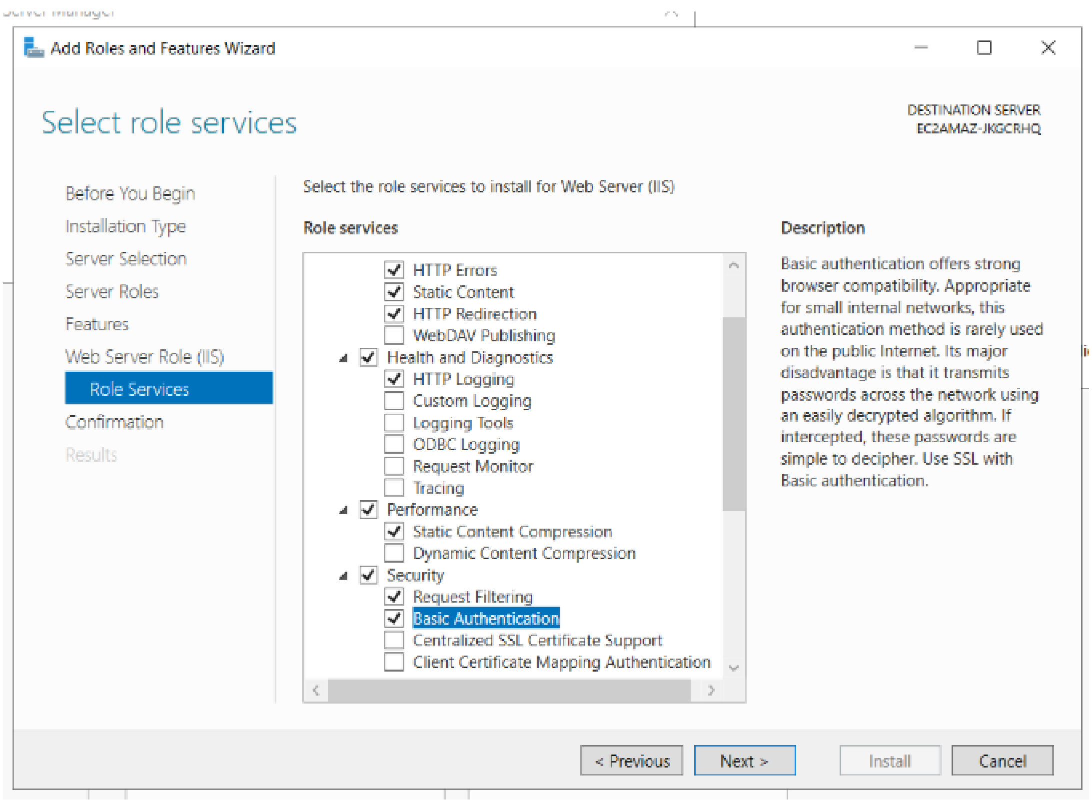
8. On the Select Features page, click **Next**.



9. On the Web Server Role (IIS) page, click **Next**.

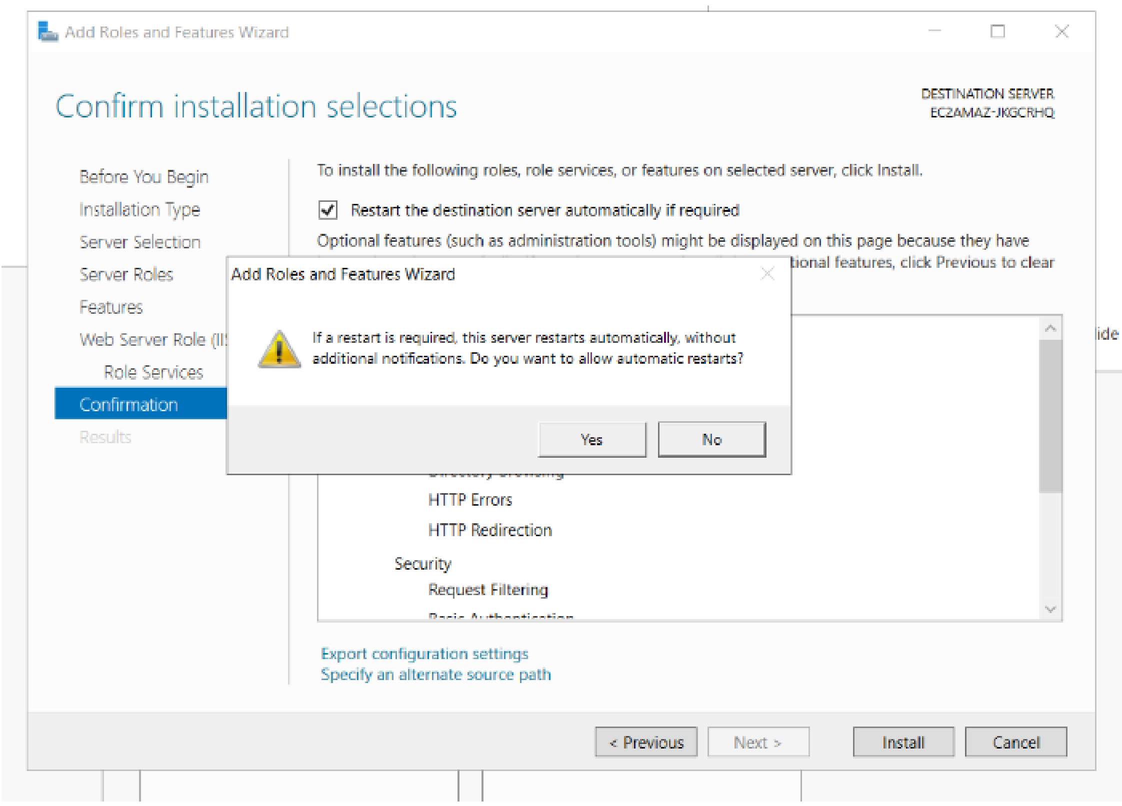


10. On the Select Role Services page, select 'HTTP Redirection' and 'Basic Authentication', and then click **Next**.

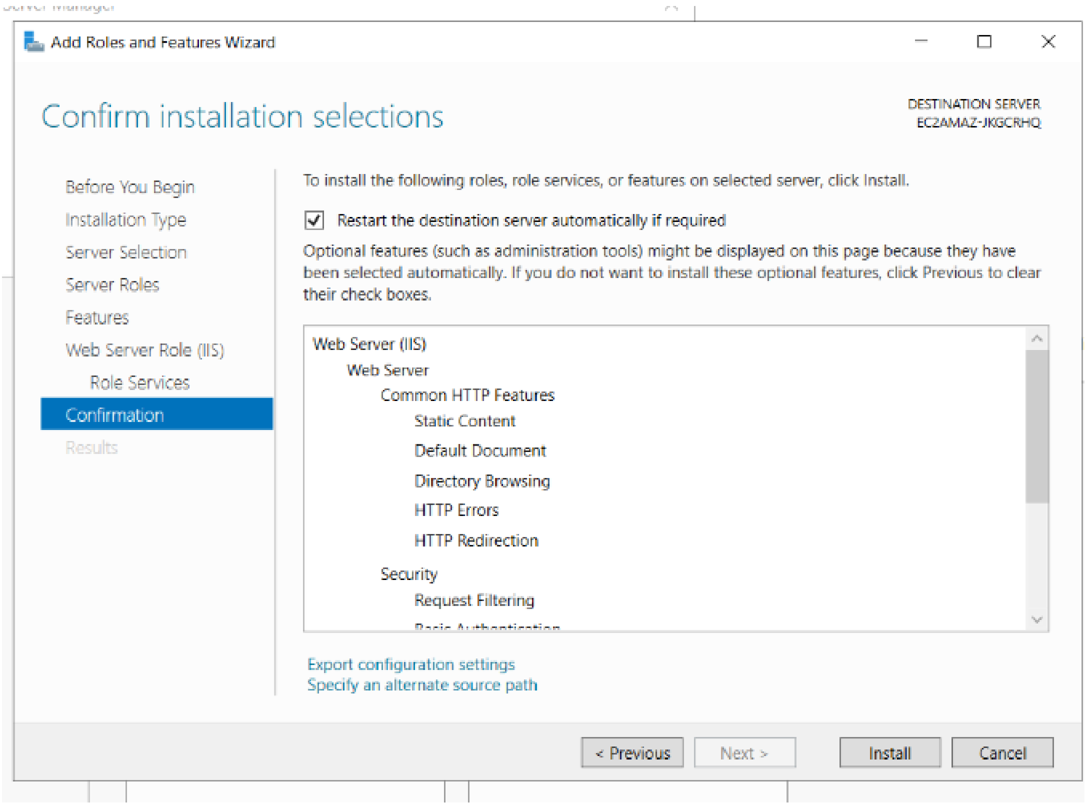


11. On the Confirm Installation Selections page, select the 'Restart the destination server automatically if required' check box; a message appears regarding automatic restarts.

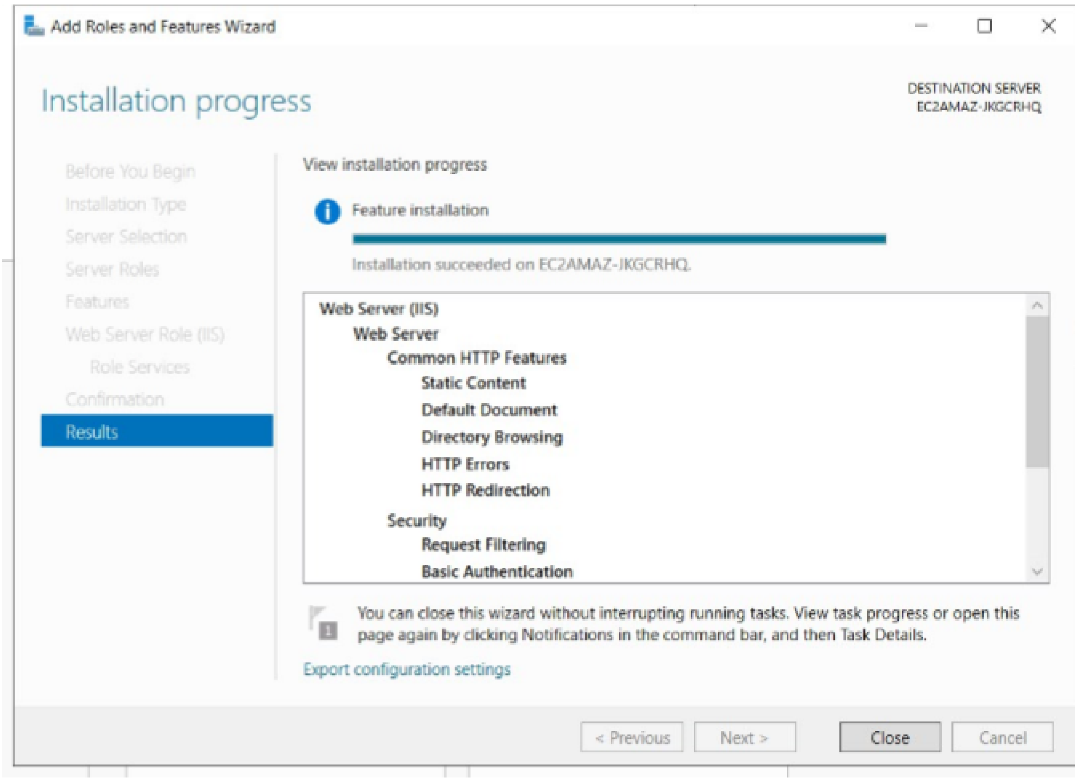
12. If a restart is required, click **Yes**.



13. Click **Install**; the Installation Progress page appears.



14. When the installation has completed, click **Close**.



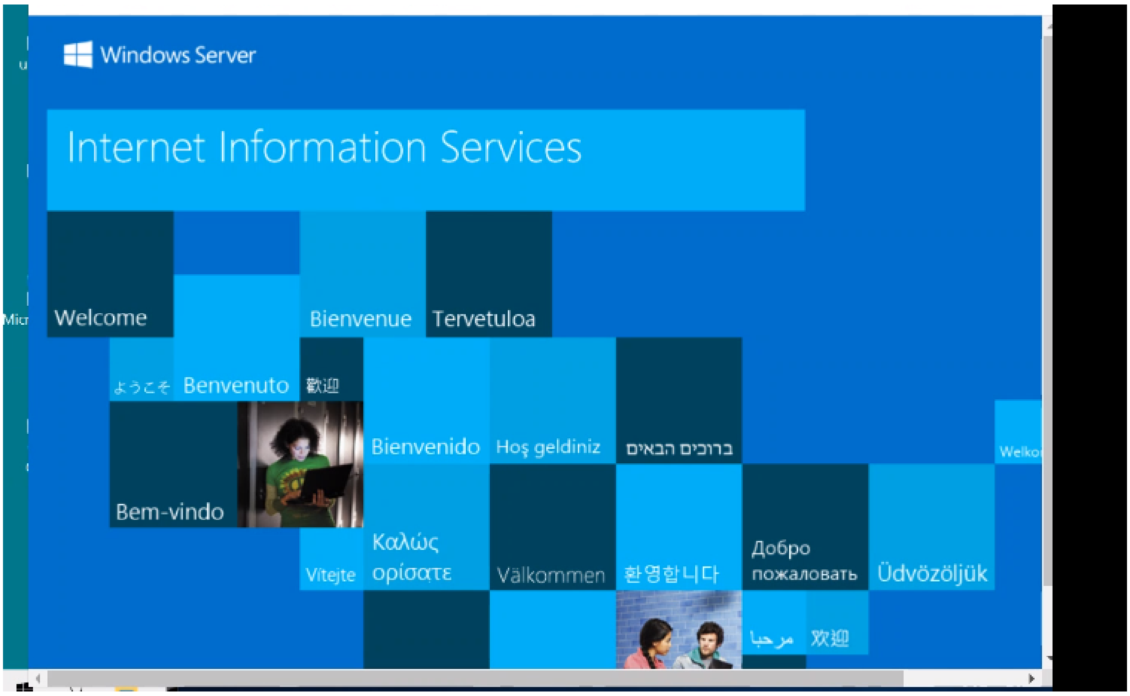
15. Close the Control Panel.

16. Using Windows Search, enter "IIS Manager".

17. Open the file location and create a desktop link to IIS Manager.

18. Open IIS Manager; ensure that your IIS HTTP server has started in the right pane (Manage Server).

19. On a different computer, try to open http://<your_site_fqdn> in the browser; the following page appears.



For more information, see the official Microsoft IIS site at https://docs.microsoft.com/en-us/iis.

Installing the SSL Certificate

There are two ways to order a security certificate:

- IIS Manager
- Open Source

Using IIS Manager

The Certificate Signed Request (CSR) can be created with the IIS Manager. This is the standard way and is described in many places. For example, go to https://www.ibm.com/support/pages/how-generate-csr-certificate-signing-request-using-iis.

Send the created CSR to the certificate authority, and then install the received signed certificates. For more information, go to https://www.ibm.com/support/pages/how-install-ssl-certificate-iis7.

Using Open Source

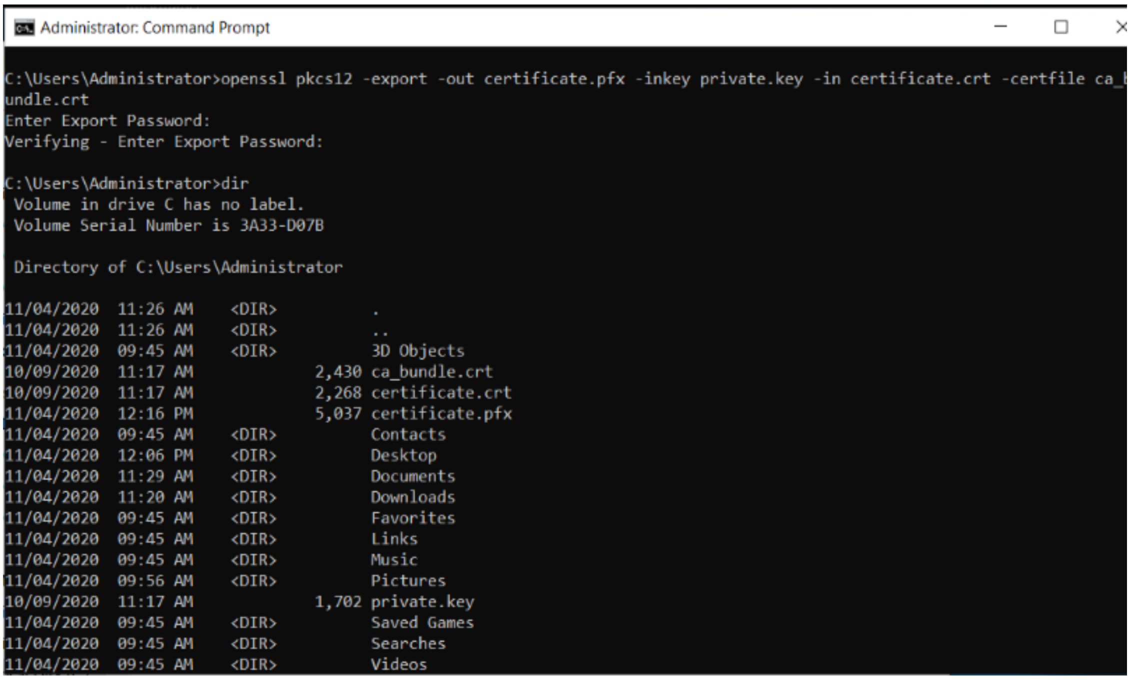
You can use the openssl open source utility, in the same way as it is used for other HTTP servers, such as Apache and Nginx. For more information, go to https://en.wikipedia.org/wiki/OpenSSL.

1. Send the CSR to the certificate authority and receive a signed certificate.
2. From the command prompt, convert the following files with the openssl utility
   - private.key
   - certificate.crt
   - ca_bundle.crt

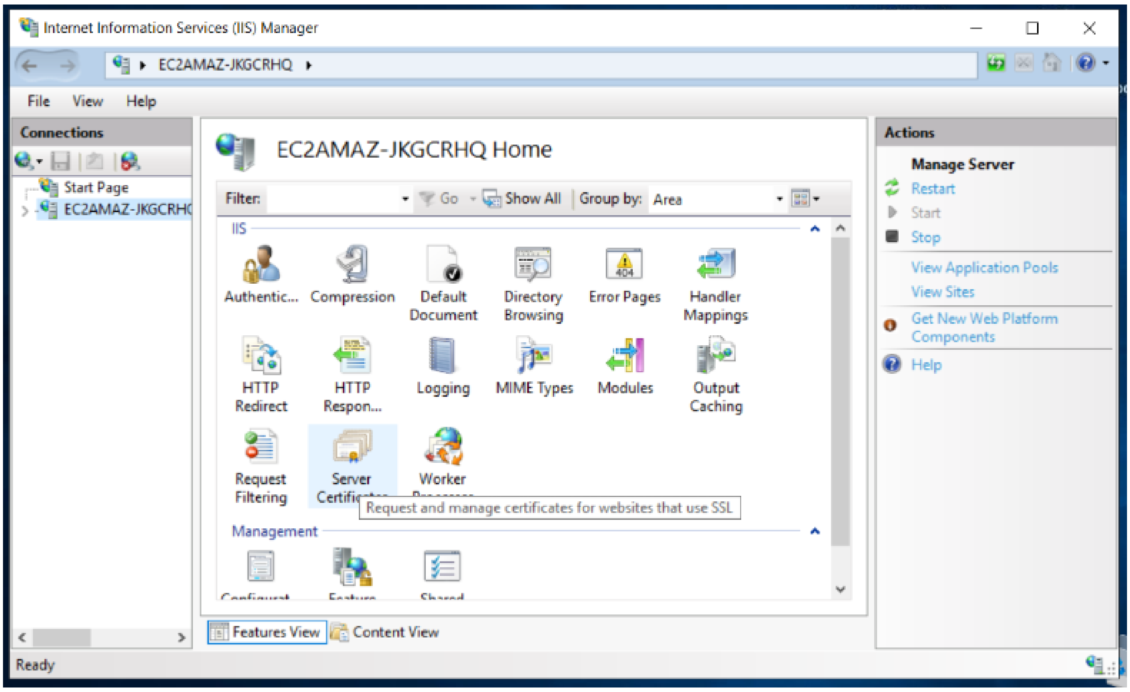
   to pfx file format using openssl.

```
openssl pkcs12 -export -out certificate.pfx -inkey private.key -in certificate.crt -certfile ca_bundle.crt
```

3. Enter and confirm the Export password.

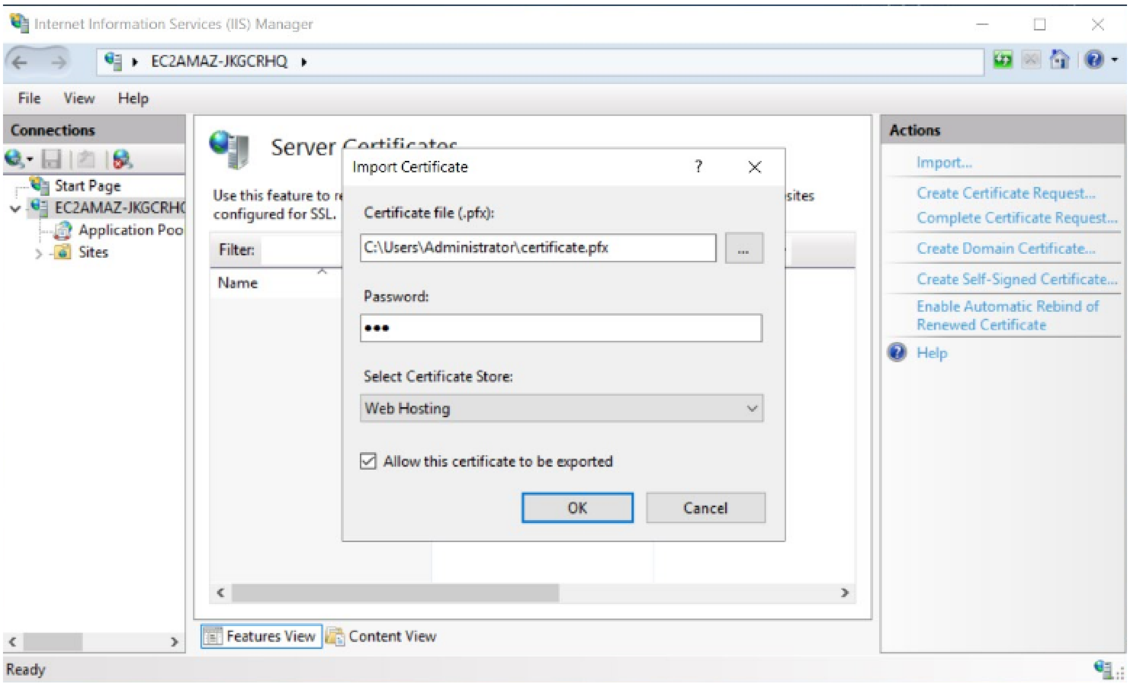


4. Create the pfx file.
5. Copy the created certificate to the Windows server using Remote Desktop Connection.
6. Open IIS Manager and from the left Connections pane, select the local host entry, below Start Page.



In the above example, the local host is EC2AMAZ-HFH46B9(EC2AMAZ-HFH46B9\Administrator. If you hover the mouse pointer on this entry, the following tooltip is shown - http://localhost.

7. Click the **Server Certificates** icon, and then from the right Actions pane, click **Import**; the Import Certificate page appears.
8. In the 'Certificate Enter' field, enter the pfx file path.
9. In the 'Password' field, enter the Export password.
10. From the 'Select Certificate Store' drop-down list, select **Web Hosting**, and then click **OK**.



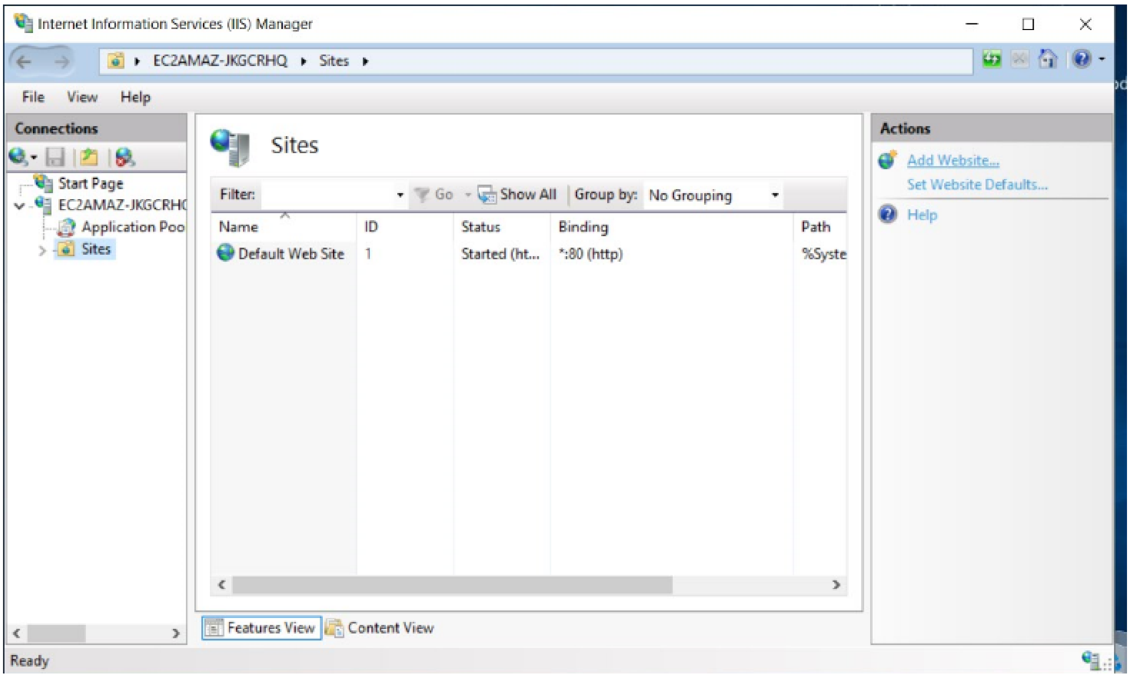
The certificate.pfx file has been installed on the IIS Manager.

Creating the HTTPS Site

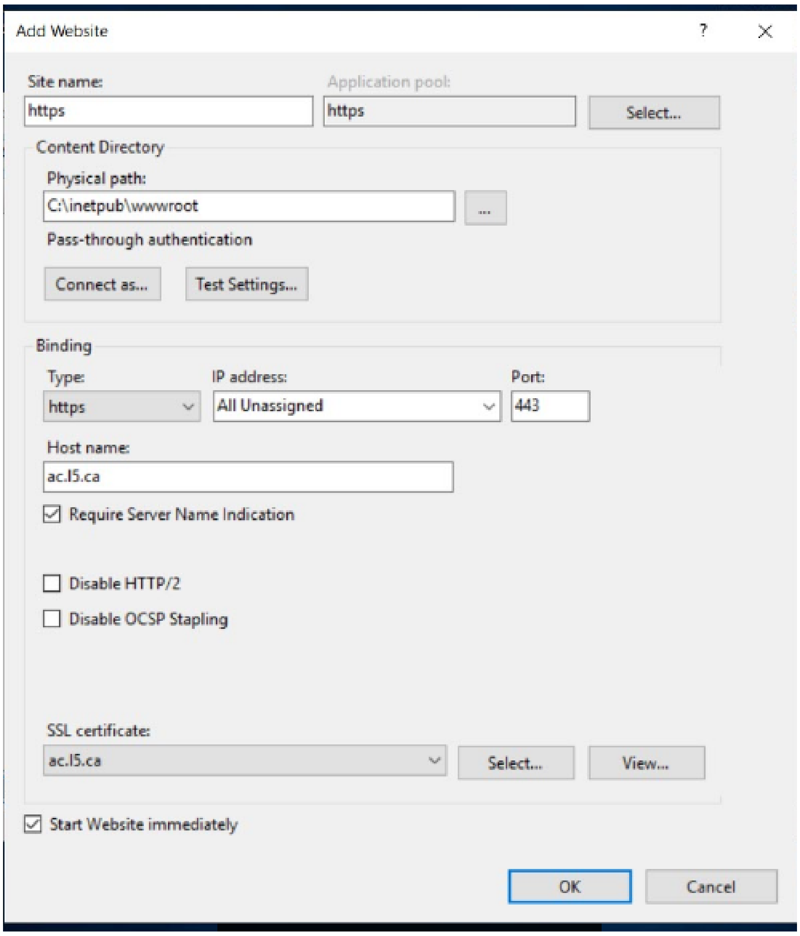
The procedure below describes how to create an HTTPS site.

➢ To create an HTTPS site:

1. Open the IIS Manager.
2. On the left Connections pane, select the local host entry under Start Page.
3. Expand Sites.



4. On the right Actions pane, click **Add Website**; the following page appears:



5. In the 'Site name' field, enter "https". In the 'Physical path' field, enter "C:\inetpub\wwwroot" (default IIS path).

6. From the 'Binding type' drop-down list, select **https**.

7. In the 'Host name' field, enter your site FQDN.

8. Select the 'Require Server Name Indication' check box.

> The 'Require Server Name Indication' check box must be selected when you create multiple HTTPS sites in IIS Manager and use many FQDNs. For a single HTTPS site, this is optional.

9. From the SSL certificate drop-down list, select the SSL certificate you created in Using Open Source on page 16.

10. Click **OK**.

11. Check the created site by opening a browser at https://<your_site_fqdn>. You should see the default IIS start page.

Configuring the HTTPS Site

To configure the HTTPS site, do the following:

- Install URL Rewriter 2.1 Component

- Create HTTP to HTTPS Request Redirection
- Configure Compression of Responses
- Enable Special Signs in Filenames

> Previously, the Web Platform Installer was used to install URL Rewriter. It's no longer supported as described in https://blogs.iis.net/iisteam/web-platform-installer-end-of-support-feed.
> Now we download the *url- rewrite module* from https://www.iis.net/downloads/microsoft/url-rewrite.

Installing URL Rewriter 2.1

The procedure below describes how to install URL Rewriter Version 2.1.

➢ **To install URL Rewriter 2.1:**

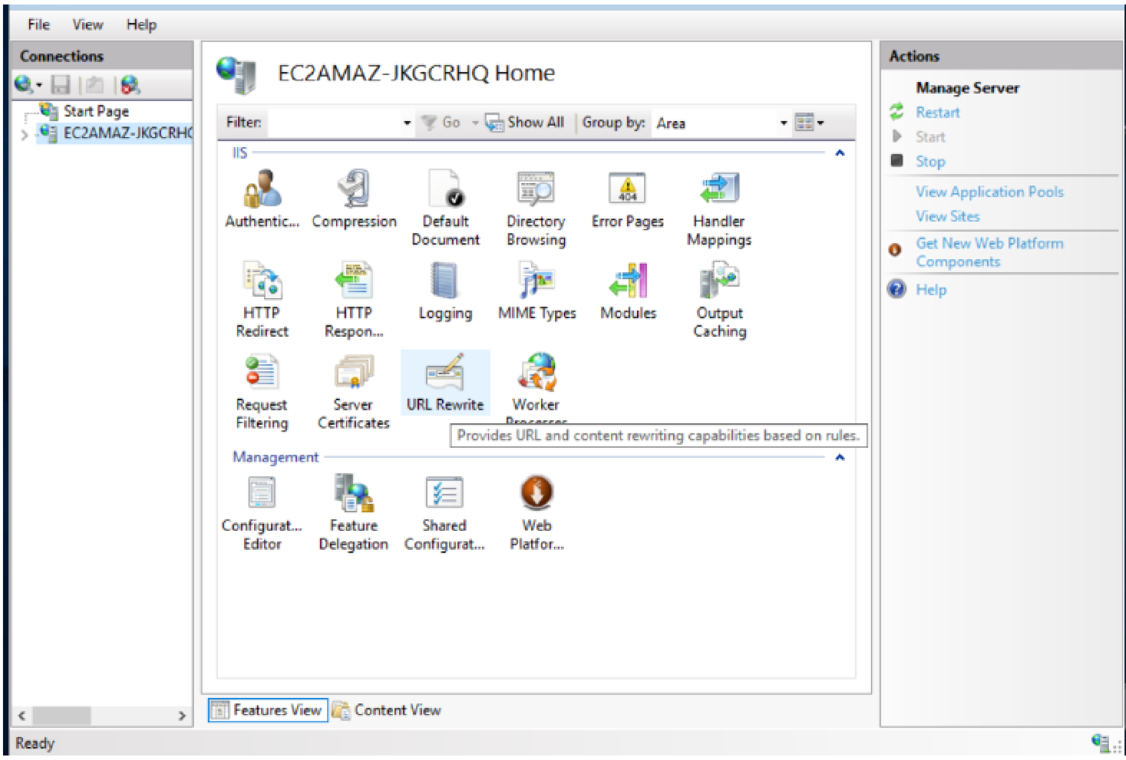
1. Download the installer by clicking here.
2. Select **Download URL Rewrite Module 2.1** x64 installer.
3. Download and open the *rewrite_amd64_en-US.msi* file.

Creating HTTP to HTTPS Request Redirection

The procedure below describes how to create HTTP to HTTPS request redirection.

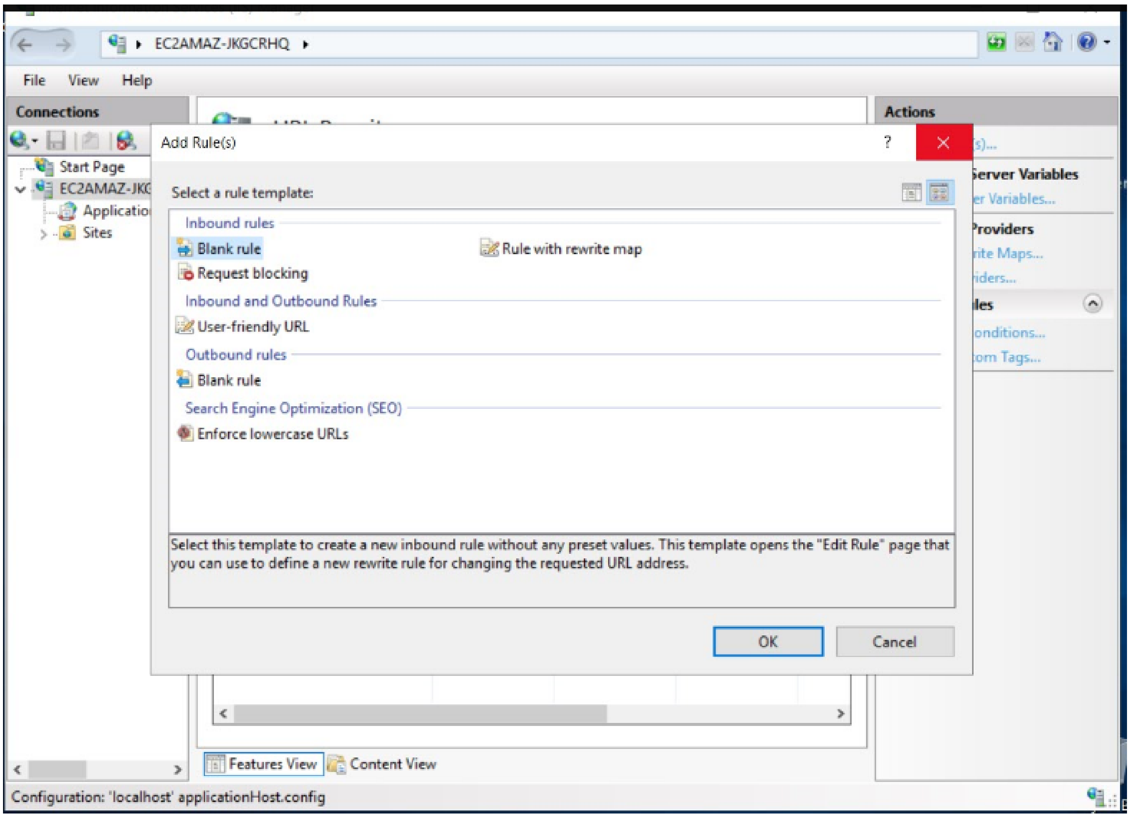
➢ **To create HTTP to HTTPS request redirection:**

1. Start IIS Manager.
2. On the left Connections pane, select the local host entry under Start Page.

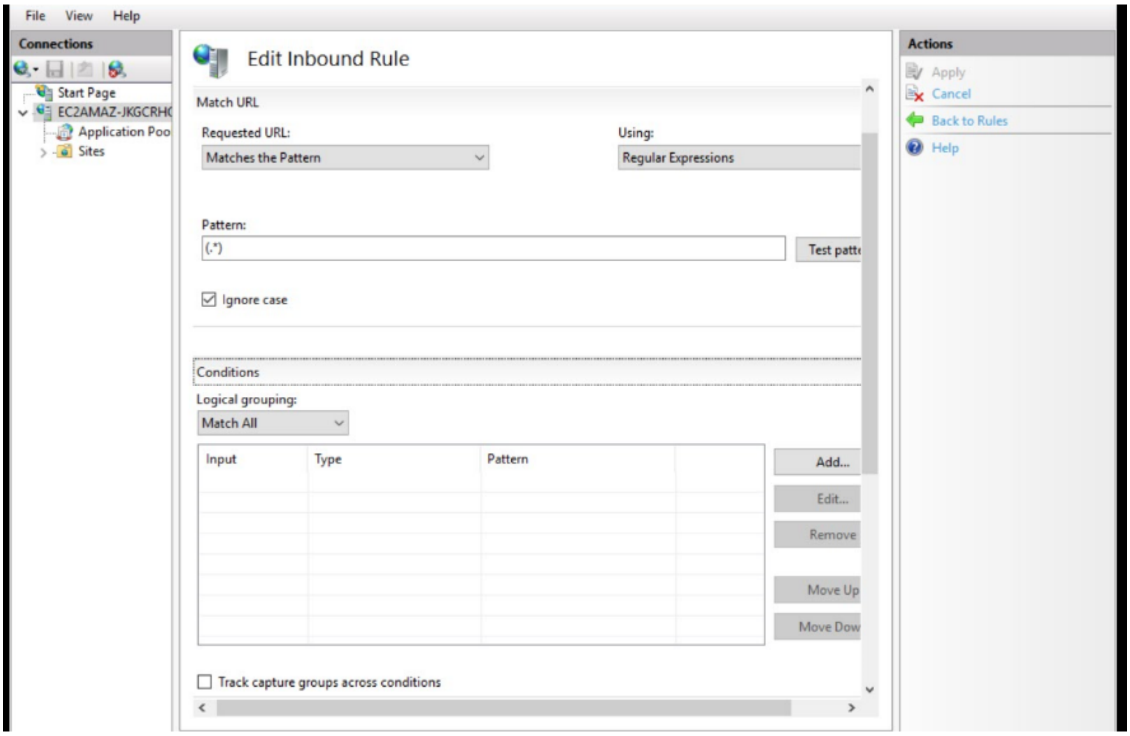


3. Click the **URL Rewrite** icon; the URL Write page appears:

4. On the right Actions pane, click **Add Rule(s)**; the Add Rule(s) page opens.



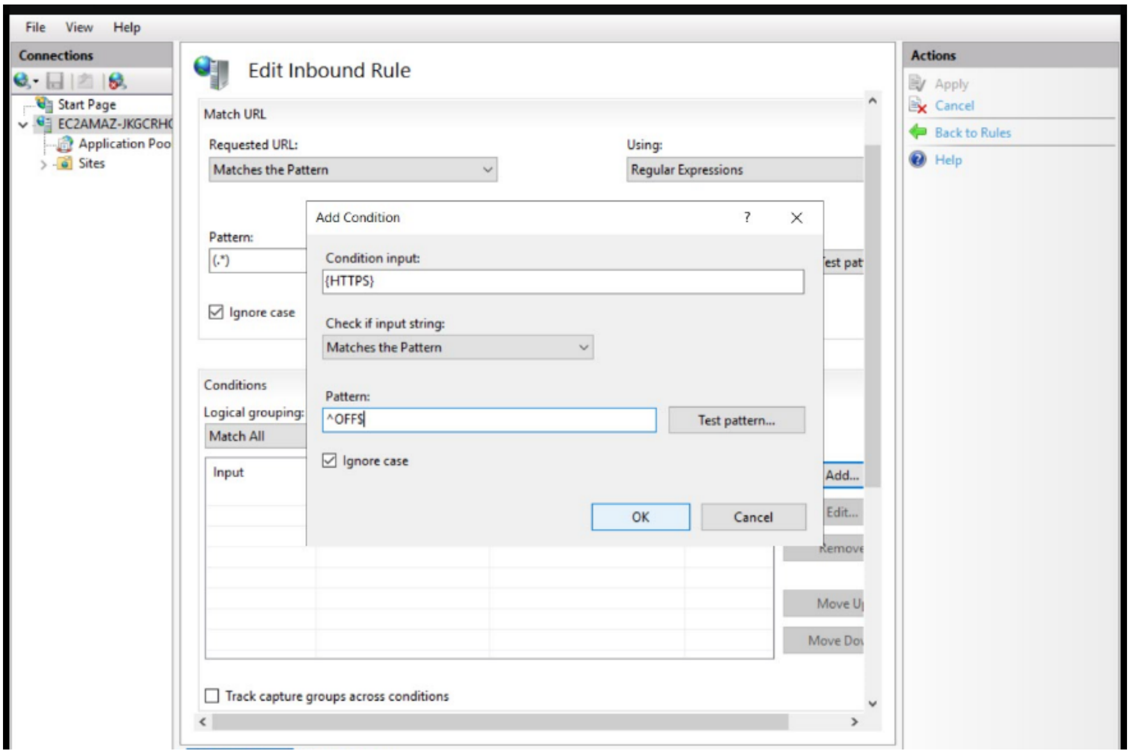
5. Select Blank Rule, and then click **OK**; the Edit Inbound Rule page appears:



6. From the 'Requested URL' drop-down list, select **Matches the Pattern**.
7. From the 'Using' drop-down list, select **Regular Expressions**.
8. In the 'Pattern' field, enter "(.*)".

(.*) is entered in the 'Pattern' field.

9. Select the Ignore case check box.
10. From the Logical Grouping drop-down list, select **Match all**, and then click **Add**; the Add Condition page appears.



11. In the 'Condition input' field, enter "{HTTPS}".

{HTTPS} is entered in the 'Condition input' field.

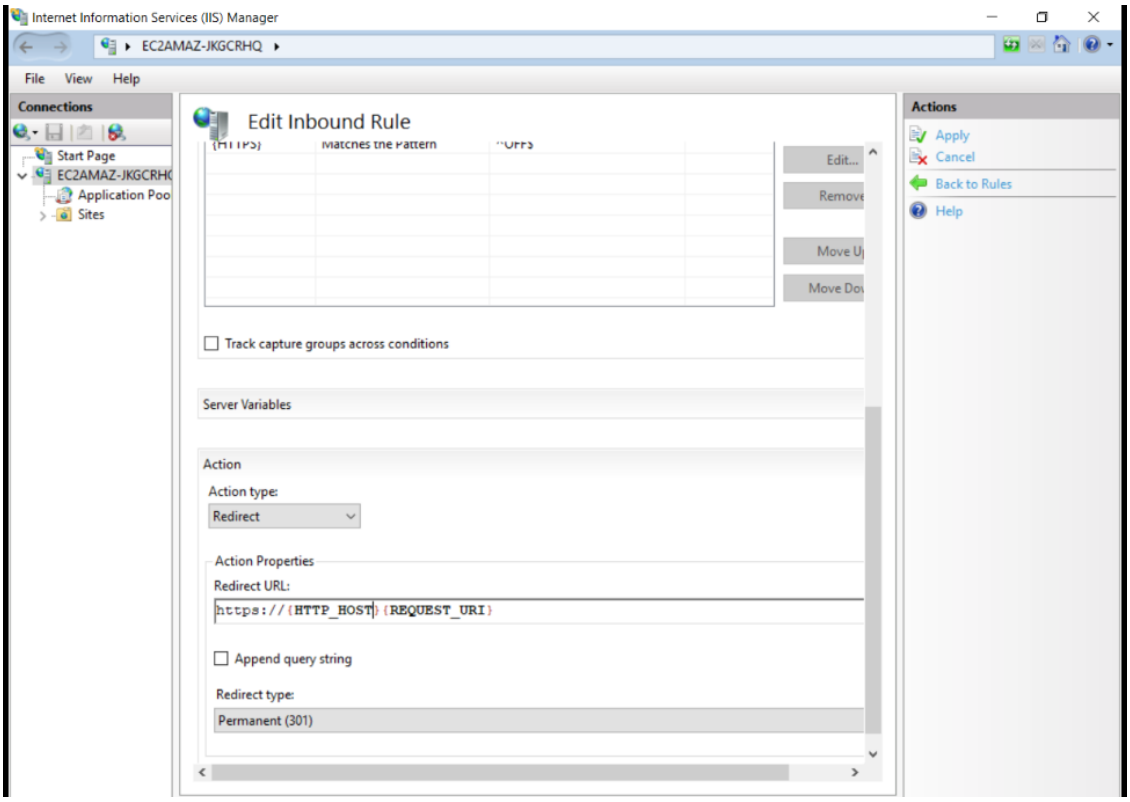
12. From the 'Check if input string' drop-down list, select **Matched the Pattern**.

13. In the 'Pattern' field, enter "^OFF$".

^OFF$ is entered in the 'Pattern' field.

14. Select the 'Ignore case' check box.

15. Click **OK**; the Edit Inbound Rule page with Action settings appears:



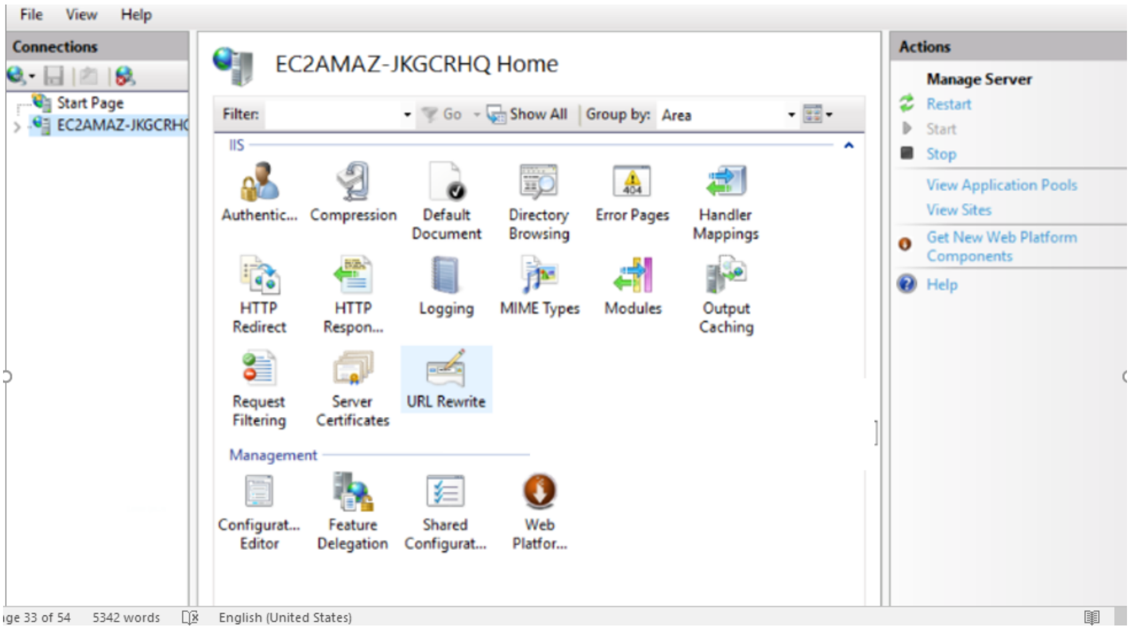
16. From the 'Action type' drop-down list, select **Redirect**.
17. In the 'Redirect URL' field, enter "https://{HTTP_HOST}{REQUEST_URI}".
18. Clear the 'Append query string' check box.
19. In the 'Redirect type' field, enter "Permanent (301)".
20. In the right pane, click **Apply**.
21. Open your browser and confirm that it has been redirected to http://<your_site_fqdn>.

Configuring Compression of Responses

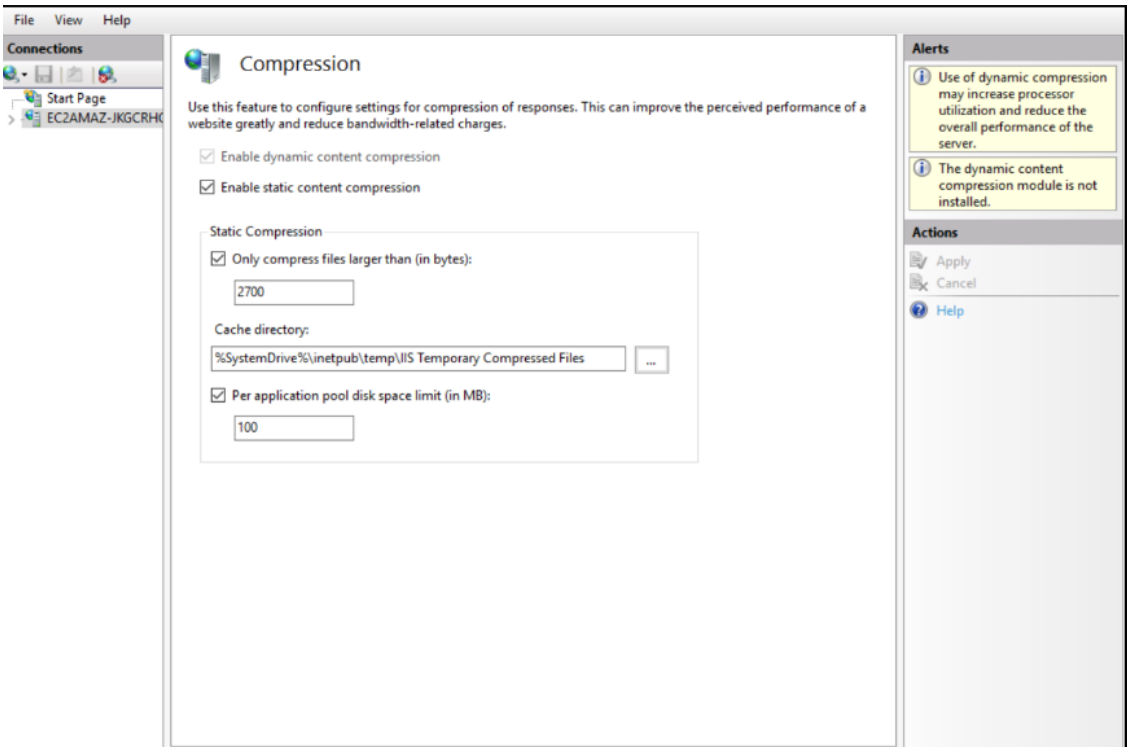
The procedure below describes how to configure the settings compression of responses.

➢ **To use compression:**

1. Start IIS Manager.
2. On the left Connections pane, select the local host entry under Start Page.



3. Click the **Compression** icon; the Compression page appears:



4. Ensure that Static Compression is enabled.

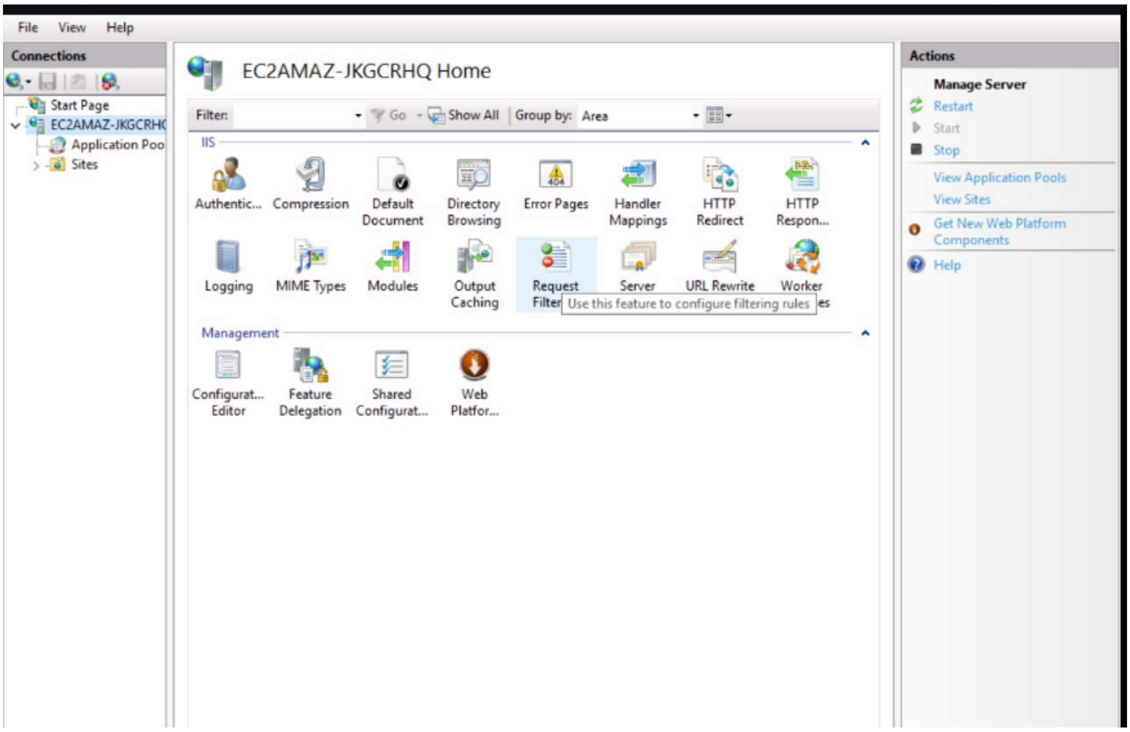
Enable Special Signs in Filenames

The procedure below describes how to enable special signs, (e.g.,+) in filenames.

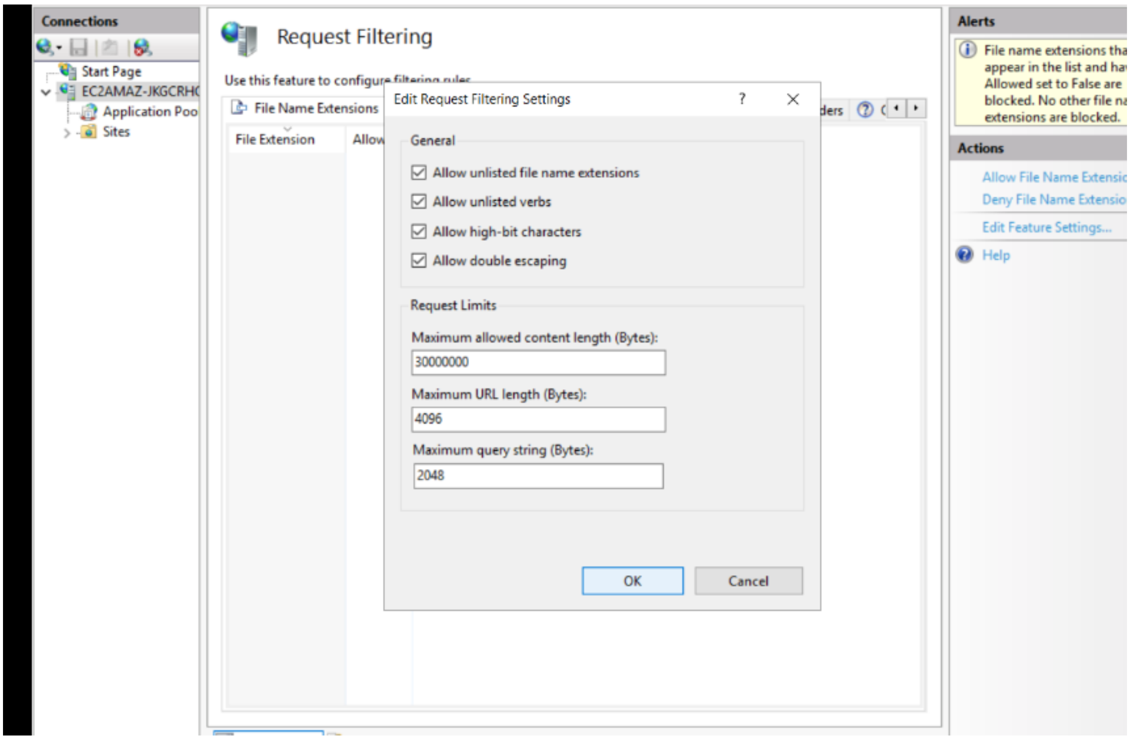
➢ To enable special signs:

1. Start IIS Manager.

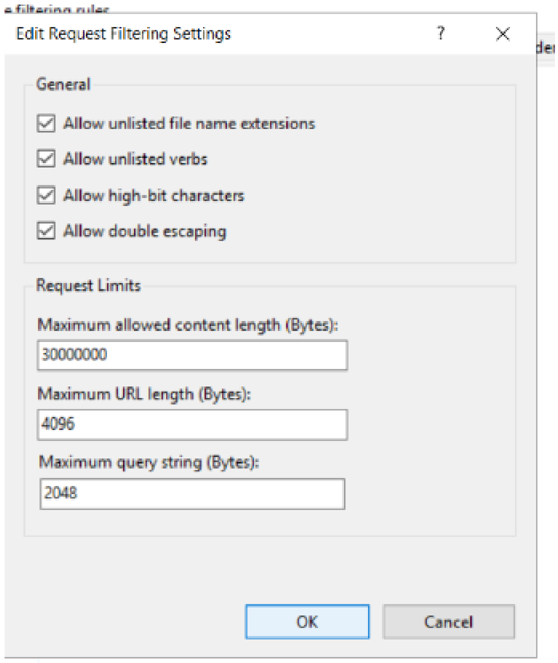
2. On the left Connections pane, select the local host entry under Start Page.



3. Click the **Request filtering** icon; the Request Filtering page appears:



4. On the right pane, click **Edit Feature Settings**; the Edit Request Filtering Settings page appears.



5. Enable the allow double escaping check box.

6. Click **OK**.

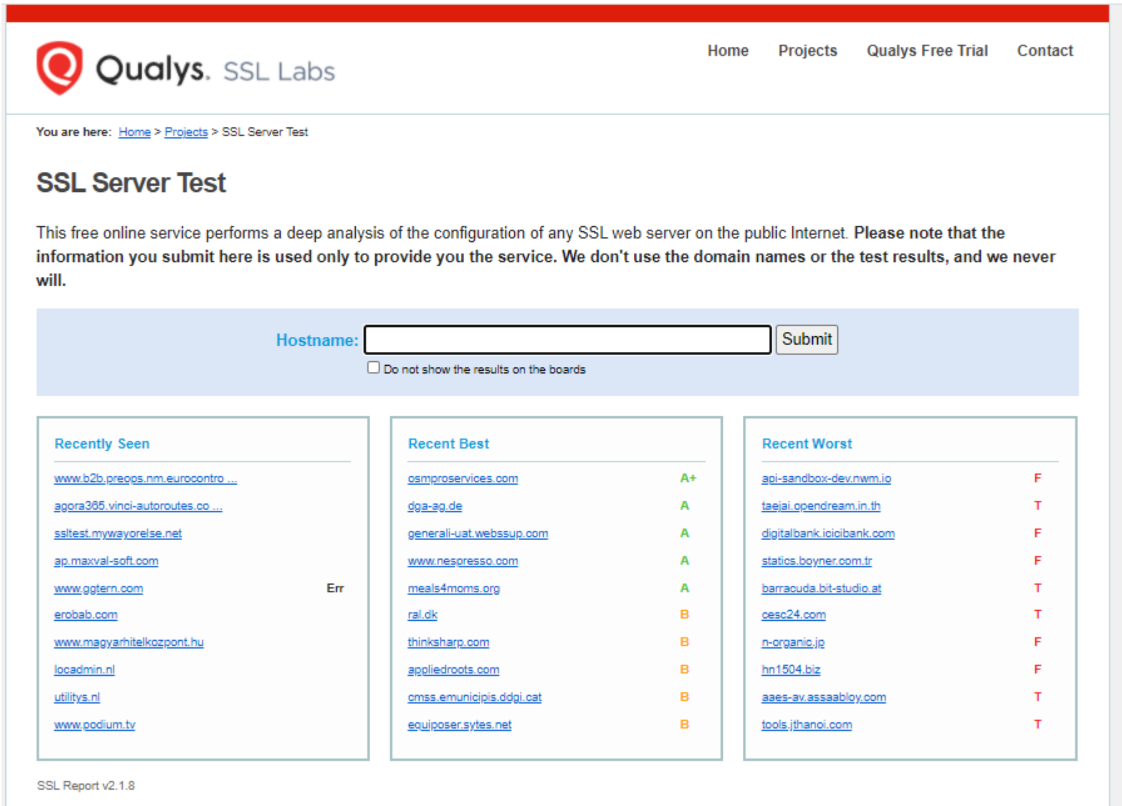
Configuring Windows Server to Only Use TLS 1.2

The procedure below describes how to configure the Windows server to only use TLS 1.2.

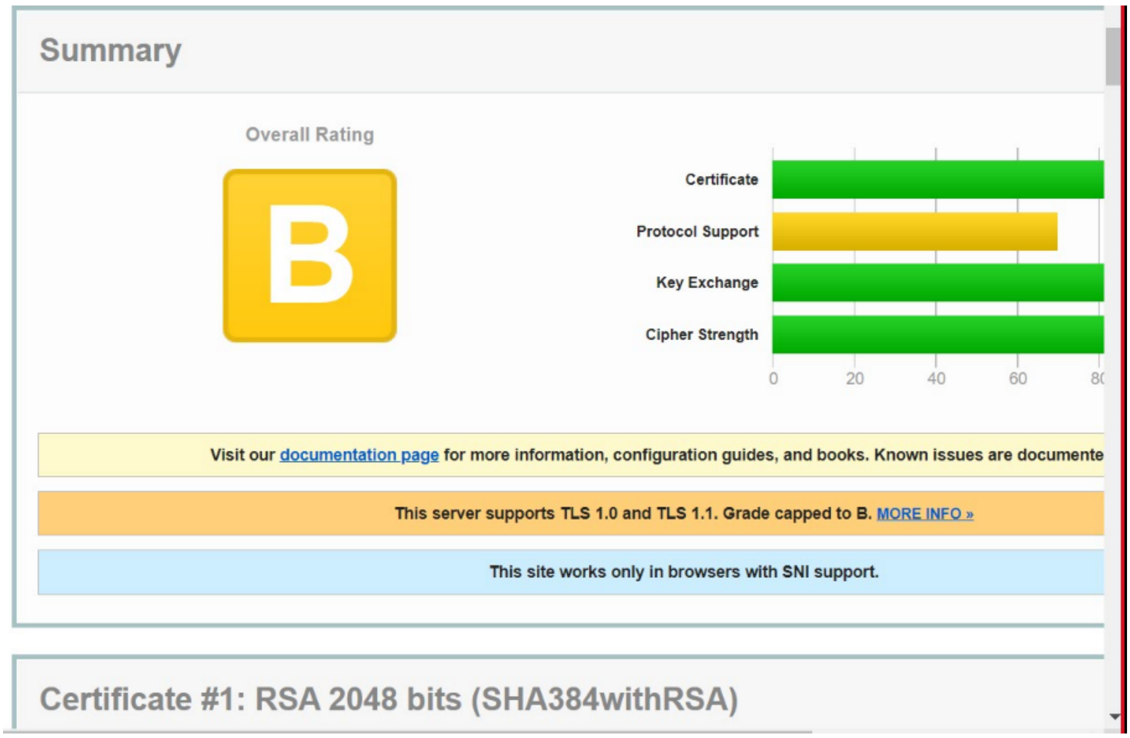
This procedure is not necessary for Windows Server 2022.

➢ To configure the Windows server to only use TLS 1.2:

1. Check the server security grade by service. Go to https://www.ssllabs.com/ssltest/index.html; the following page appears:



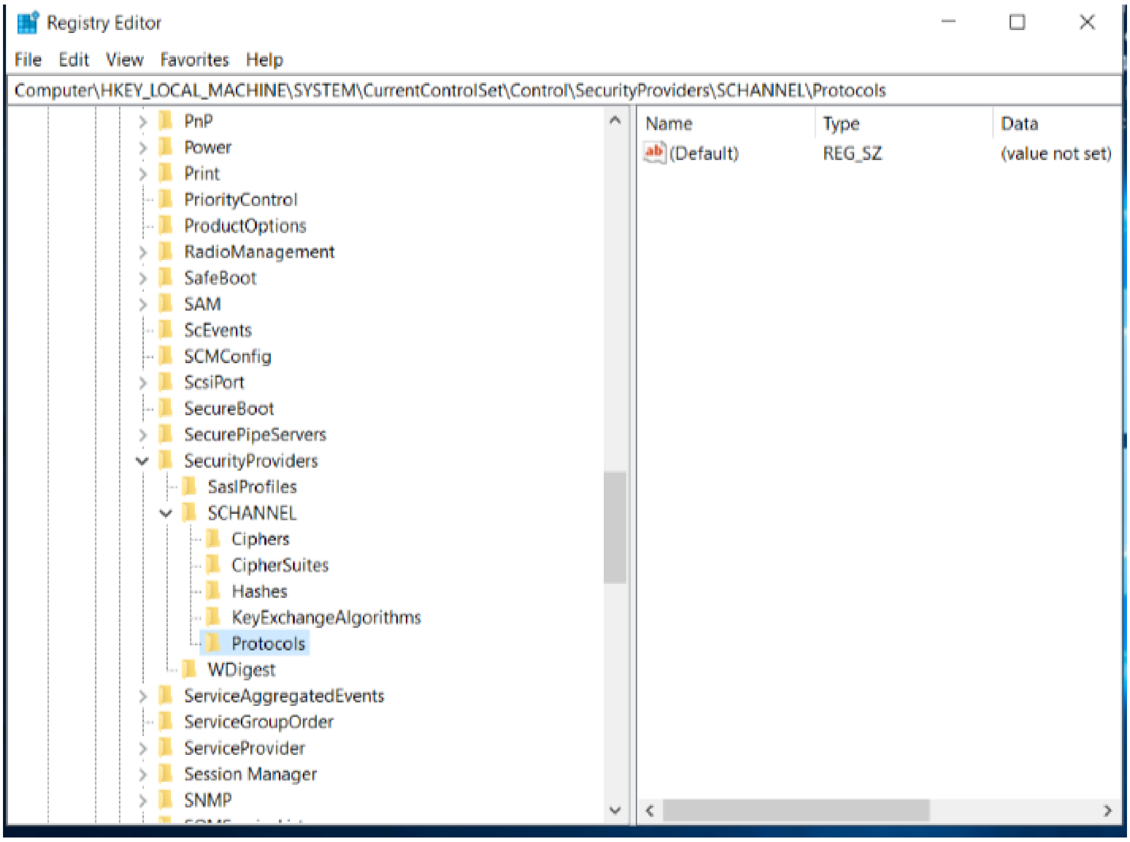
2. Enter the hostname, and then click **Submit**; a summary report is displayed with the following message: This server supports TLS 1.0 and TLS 1.1. Grade capped to B.



3. Disable TLS 1.0 and TLS 1.1 in the Windows server (refer to https://docs.microsoft.com/en-us/windows-server/security/tls/tls-registry-settings).

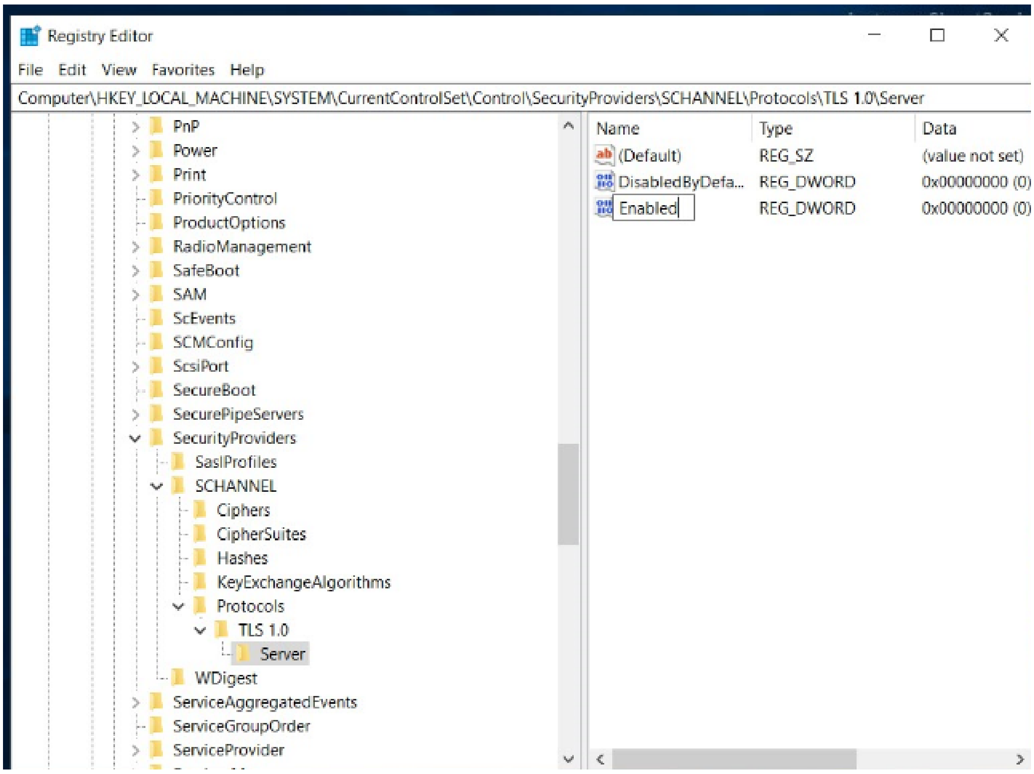
4. From the Windows start menu, open the Registry Editor. Run as the administrator.

5. Open Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols



In Windows Server 2019, there are no “TLS 1.0” and “TLS 1.1” keys; there is only a Default value.

6. Add the TLS 1.0 sub-tree, by right-clicking the Protocols entry.
7. Select “New” “Key”, and then create the TLS 1.0 key.
8. Right-click the TLS 1.0 entry, and then select “New” “Key”.
9. Create the Server key.



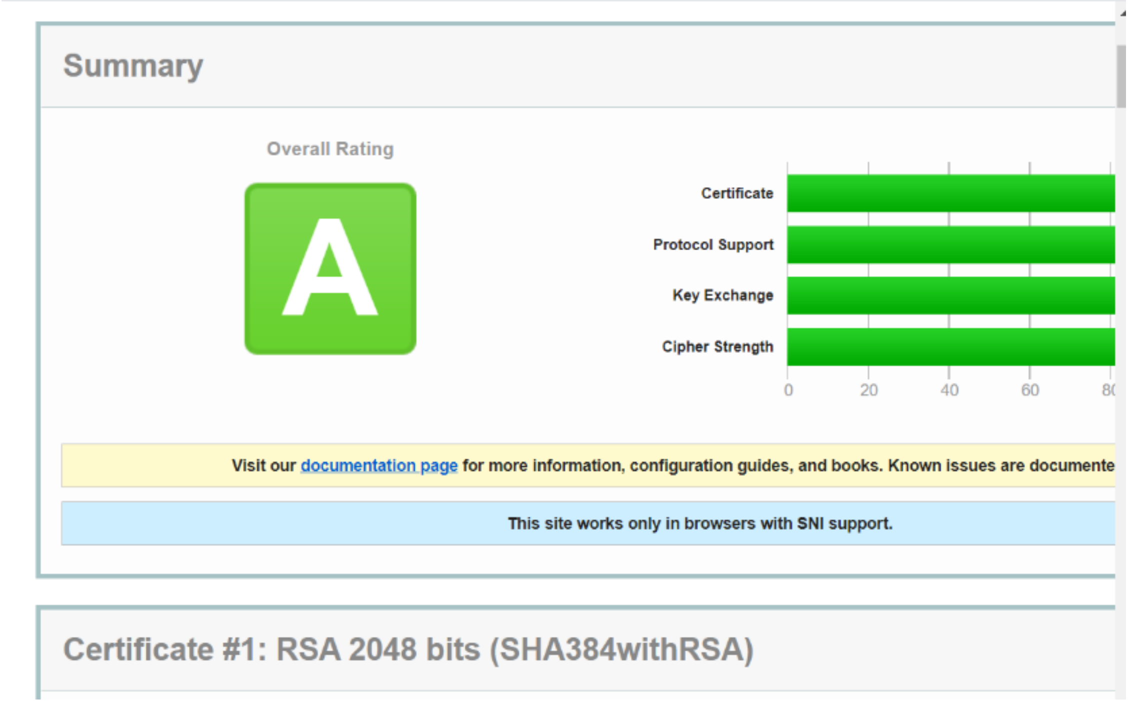
10. Right-click the Server entry, select “New” “DWORD (32-bit) Value” and then create the DisabledByDefault value.
11. Right-click the Server entry, select “New” “DWORD (32-bit) Value”, and then create the Enabled value.

> Both DisabledByDefault and Enabled should be set to 0.

12. Add the TLS 1.1 sub-tree in the same way as you did for the TLS 1.0 sub-tree. See Step Add the TLS 1.0 sub-tree, by right-clicking the Protocols entry. on the previous page.
13. Right-click the 'Protocols' entry, select “New” “Key”, and then create the TLS 1.1 key.
14. Right-click the TLS 1.1 entry, select “New” “Key”, and then create the Server key.
15. Right click the Server entry, select “New” “DWORD (32-bit) Value”, and then create the DisabledByDefault value.
16. Right-click the Server entry, select “New” “DWORD (32-bit) Value” and then create the Enabled value.

> Both DisabledByDefault and Enabled should be set to 0.

17. Reload Windows Server 2019.
18. Re-check the security grade, by opening https://www.ssllabs.com/ssltest, and then clicking the **Clear cache** button; the result will now be Grade A.



Deploying WebRTC Client on IIS

➢ **To deploy on IIS server:**

1. Unzip the file webrtc-web-client-vx.x.x.zip.
2. Copy the unzipped content to the *C:\inetpub\wwwroot\webrtc_client* directory.

Upgrading WebRTC Client on IIS

➢ **To upgrade the WebRTC client on IIS:**

To upgrade the WebRTC client on IIS, see Upgrading the Web Client to a New Release.

Customizing WebRTC Client

The procedures below describe how to customize the WebRTC client.

Advanced Options Configuration File

The advancedOptions.js file allows you to edit several attributes. Most of the configuration is self- explanatory and should remain as default configuration. Below is a list of the most important parameters:

- **defaultServerConfig:** Defines the default SIP server configuration.
  - **sipDomain:** Defines the domain name (for example, audiocodes.com) used by the WebRTC client in the SIP messages (INVITE/REGISTER) sent to the WebRTC gateway.

- **sipServerAddress:** Defines an array of the default SBC addresses list, shown on the client server field (for example, wss://s-bcGENLab1.customers.audiocodesaas.com:10081). If a server's URL in the list is not responding, another one in the list will be attempted. The 'prioritize' value below determines how the next URL is chosen.
- **prioritize:** If 'true', then the SBC URLs list is sorted by priority, so that if connecting to a URL fails, the next one in the list is attempted. If 'false', then the next attempt is selected randomly.
- **sipIceServers:** Recommended to leave empty but if required may be edited to contain list of ice servers as shown below.
- **backupSipServerAddress:** Defines an array of the backup SBC addresses list (same structure as sipServerAddress). If the address is set, the dual registration feature is enabled.
  Default: empty list, dual registration feature is disabled.
- **guiEnabled:** Defines whether to allow showing server configuration settings in the client GUI. Default: true.
- **disableEditing**: Defines whether modifying server configuration by the user through the Settings UI is disabled, so that server configuration settings are always determined by the values from the configuration file. Default: false (the user is allowed to edit).

- **storageConfig:** Configure the client storage and caching behavior.
  - **location:** Defines the web storage mechanism the client uses to cache user data, including authentication states. Possible values:
    - **localStorage:** Uses the browser's persistent local storage.
    - **sessionStorage:** Uses only the session cache (e.g., tab lifetime).
- **defaultOAuthConfig: DEPRECATED, see "Migrating Authentication Providers and Contact Center Configuration from Previous Versions".**
- **ACDClientConfig:** Configuration for contact-center agent ACD support.
  - **enabled:** Defines whether Automatic Call Distribution is enabled.
  - **notReadyDefaultReasonCode:** Defines the value that is used by default, when the client automatically goes to a "not-ready" state. It can be one of the list in notReadyReasonCodeMapping, or a totally different one.
  - **notReadyReasonCodeMapping:** Defines the list of codes and corresponding text values that are presented to the user as selectable not-ready reason codes.
  - **notReadyStatusValue: DEPRECATED.** Use *notReadyDefaultReasonCode* and *notReadyReasonCodeMapping* instead.
- **sipAccountGUIConfig:** Configures the GUI display and control of SIP account details.
  - **sipDisplayNameHidden:** Determines whether to hide SIP display-name from GUI altogether. Default: false.

- **sipDisplayNameDisabled:** Determines whether to disable SIP display-name configuration in Settings. Default: false.

- **notificationOptions:** Optional configuration for displaying a warning when auto-play is disabled by the browser. By default, the warning will not be displayed.
  - **focusWindowOnClick (true / false):** Determines whether to bring the browser tab into focus when clicking on browser notifications from the client. Default: true.
  - **autoplay:**
    - **showAutoplayDisabledAlert (true / false):** Upon page load, displays an alert to the user on the page itself. Default: true.
    - **showAutoplayDisabledNotification (true / false):** Upon page load, attempts to display a browser notification. This is only supported for now with Chrome, Edge and Safari. Default: true.
- **autoplayNotificationOptions: DEPRECATED**. Use **notificationOptions** > **autoplay** instead.
- **customerInfoContextDisplayMapping:** Optional key-value json object: When an incoming call INVITE message arrives with the “X-Customer-Info” header for displaying customer information, this object can map keywords from the customer information into user-readable text to display. For example: customerInfoContextDisplayMapping = {advice: "Getting expert advice for the product"}
- **isVideoEnabled:** Determines whether the client has video support.
- **enableAddVideo:** Enables adding video to audio calls.
- **isScreenSharingEnabled:** Enables / Disables the screen sharing feature.
- **isCrossScreenSharingAllowed:** When incoming screen sharing occurs during a call, the client should also allow outgoing screen sharing. Default: False.
- **isAllowedToResumeLocalHoldByRemoteReInvite:** This value indicates whether or not the client is supposed to release the local hold, when receiving a reINVITE with the sendrecv media direction. Default value: false.
- **registerExpires:** Defines the SIP registration expiry time (seconds)
- **voiceQualityMonitorEnabled:** Enables / Disables voice quality monitoring.
- **restoreCallQualityMaxDelay:** Defines the maximum interval to store the last call quality score.
- **restoreCall:** Determines whether call restoration functionality is enabled.
  - True - upon refresh the client restores the previous call.
  - False - call won't restore upon refresh.
- **restoreCallMaxDelay:** Defines the maximum interval to restore a call after page reload (in seconds).
- **reconnectIntervalMin**: Defines the minimum interval between WebSocket reconnection attempts (in seconds).

- **reconnectIntervalMax:** Defines the maximum interval between WebSocket reconnection attempts (in seconds).
- **dtmfUseWebRTC:** Determines which DTMF type to use: RFC 2833 or SIP info.
- **dtmfDuration:** Defines the duration of the DTMF tone (in milliseconds). The default value is 100.
- **dtmfInterToneGap:** Defines the interval between two DTMF tones (in milliseconds). The default value is 500.
- **useSessionTimer:** Enables Session Timers (as per RFC 4028). Default: False.
- **noAnswerTimeout:** Defines the interval in seconds, from when an incoming call is received, until it automatically terminates due to no answering. Default: 60 seconds.
- **avoidTwoWayHold:** If the call is in remote Hold, disable the local Hold button to avoid 2 way holds.
- **disableOutgoingCalls:** If set to “true”, it does not allow outgoing calls.
- **autoAnswerOptions:** Defines the options for behavior with auto-answering a call:
    - **answerDelayedOfferWithVideo:** When an incoming offer with no SDP arrives with an auto-answer trigger, answer with video media. Default: False.
    - **autoAnswerDefaultDelaySeconds:** Defines the delay in seconds before an incoming call is auto-answered, given that the call is eligible for auto answer. Default is 0 (no delay).
- **callHistoryConfig:** Defines call history configuration.
    - **maxEntries:** Defines the maximum number of call history entries per user. Default: 100
    - **guiEnabled:** Defines whether to allow the call history GUI. Default: true.
- **maxSizeForCallHistory:** Defines the maximum size for the client’s call history database.**DEPRECATED.** Use *callHistoryConfig* > *maxEntries* instead.
- **layoutConfig:** This section contains several color theme configurations.
    - **alwaysShowFullCallsList:** Defines whether or not to allow the GUI calls list to expand or collapse.
        - If false (default), allow the calls list GUI to collapse and show only the currently focused call.
        - If true, the calls list will not collapse and will always show all existing calls.
- **supportedAuthenticationSchemes: DEPRECATED, see “Migrating Authentication Providers and Contact Center Configuration from Previous Versions”.**

> Auto-login from within a CTI software (e.g., Genesys WDE):
> When embedding the client in a program such as Genesys Workspace Desktop Edition, auto-login works if “user-password” is the only available scheme.

> Debug mode is used for debugging purposes only and should not be set on customer deployments.

- **modes:** Adds miscellaneous patches to the SDK for various features:
  - **cache_register_auth_mode:** reuse SIP Authorization header for REGISTER refresh requests, so that the user agent will not have to be challenged for every REGISTER refresh. Default value: true.
  - **ice_timeout_fix:** timeout interval in milliseconds for ICE candidate gathering. Default value: 2000 ms.
- **VDIIntegrationConfig:** Defines configuration for integrating with VDI (Virtual Desktop Infrastructure) solutions:
  - **vdiSolutionType:** The VDI solution to be used. Supported values:
    - “none” for no VDI integration (default value)
    - “citrix” for Citrix Workspace VDI

> To work with the Citrix VDI, the Citrix remote machine must be configured to enable proper client functionality. See Configuring Citrix Remote Desktop for Integration with WebRTC Client on page 58 for details.

- **mediaDeviceSettings:** Configuration for media device selection.
  - **deviceCategories:** List of allowed media device categories. Defaults to ["microphone", "voiceoutput", "ringeroutput", "camera"]. Possible values:
    - **microphone:** WebRTC voice input device
    - **voiceoutput:** WebRTC voice and audio playback output device
    - **ringeroutput:** Secondary output for ringing sound of incoming call to play simultaneously with voiceoutput
    - **camera:** WebRTC video input device
  - **allowMediaDeviceSelection:** If set to “true”, it allows the user to select the media input / output devices to use for WebRTC voice in calls. Default value: false.
  - **devicesMustBeSelectedByUser:** When media device selection is allowed, and the user has not selected audio input / output devices; If “false”, the default media input / output devices are automatically selected. If “true”, the user is prompted to select media devices explicitly. Default value: false.

> For the Citrix VDI solution (vdiSolutionType = "citrix"), both values are overridden and considered to be "true".

- **codecFilter:** Control codec behavior: Filter and set priority for WebRTC SDP codec generation.
    - **audio:** Control audio SDP codec generation.
        - **remove:** Array of codec names to remove (e.g., [isac, 'pcma',]).
        - **priority:** Array of codec names prioritized in descending order, so that SDP creation will arrange them such (e.g., ['opus', 'pcma', 'pcmu']).
    - **video:**
        - **remove:** Same as "audio", but for video codecs.
        - **priority:** Same as "audio", but for video codecs.
- **localVideoFilter:** Client configuration for camera-capture video filters such as virtual background.
    - **supported: DEPRECATED**. Use *virualBackgroundSupported* instead.
    - **virualBackgroundSupported:** If set to "true", the client includes settings for the user to apply virtual background effects. If set to "false", these settings are not available to the user
    - **maxSizeForLocalBGImages:** Defines the maximum additional amount of possible virtual background images that the user can choose in Virtual Background Settings.
- **localizationConfig:** Configures the client usage of its text resource file. See the Language and Text Customization section for text customization.
    - **defaultLanguageResrouceURL:** Sets an alternative location for the text resource. See the Language and Text Customization section.
    - **waitForDefaultResourceFetch:** Defines whether to hide the GUI until the text resource is downloaded. See the Language and Text Customization section.
- **customGeneralSIPHeaders:** Defines optional custom SIP headers that are to be added to outgoing SIP requests and responses. Defined as an array in the following format: ["headerName1: headerValue1", "headerName2: headerValue2", ...]
    - Currently the custom SIP headers are only added to: REGISTER, INVITE, re-INVITE to add video, INFO except DTMF, call answer and call reject.
- **isAgentAssistEnabled:** Whether or not to enable Agent Assist (Default = False).
    - Show/Hide the Agent Assist chatbot UI during a call
    - Show/Hide the Agent Assist settings
    - Show/Hide Agent Assist chat history for recent calls.

- **agentAssistDefaultBotName:** The default bot name that the client requests to activate for Agent Assist.
- **agentAssistDefaultServiceUrl:** The default URL for the machine running the Agent Assist service (typically part of the Voice-AI Connect server), to which Agent Assist requests are sent.
- **agentAssistDefaultServicePath:** A socket-io service path, used as a URL suffix for the service URL.

> It must be identical to the service-path configured in the server.

- **agentAssistMaxHistoryEntries:** The number of recent agent assist chat sessions that can be stored (Default = 50).
- **agentAssistDefaultConversationConfig:** Configuration related to conversation parameters defined here.
  - **defaultRoles:** Default participant names of the conversation, to distinguish which party is speaking when a call-transcript message arrives:
    - **caller:** Default participant name of the caller. Typically, “participant”.
    - **callee:** Default participant name of the callee. Typically, “participant-2”.

Depending on deployment platform, this file is located in:

- **Apache server:** */var/www/html/webrtc_client/static/js/advancedOptions.js*
- **NGINX server:** */var/html/webrtc_client/static/js/advancedOptions.js*

```
var defaultServerConfig = {
  sipDomain: "audiocodes.com",
  sipServerAddress: ["wss://sbcGENLab1.customers.audiocodesaas.com:10081"],
 sipIceServers: [“74.125.140.127:19302”,”74.125.143.127:19302”]
};
```

Configure Integrations with Contact Center Platforms

The advancedOptions.js file allows you to set the contactCenterConfig object, which defines integration with Contact Center platforms, such as Genesys Pure Engage / PureCloud, so that the web client functionality can be seamlessly integrated into these systems.

For example:

```
var contactCenterConfig = {
  integrations: [
    {
      enabled: true,
      contactCenterPlatform: "genesys_pure_engage",
      serviceName: "Genesys Pure Engage",
      customSipHeaderNames: {
        contactCenterCallUuidHeader: "x-genesys-calluuid",
      },
    },
    {
      enabled: true,
      contactCenterPlatform: "genesys_cloud",
      serviceName: "Genesys Cloud US-West-2",
      environment: "usw2.pure.cloud",
      clientId: "8612e5af-3c70-4178-ba3e-4333dcee0b32",
      redirectUri: "",  // Empty for current, or for specific URL
      customSipHeaderNames: {
        contactCenterCallUuidHeader: "x-inin-cnv",
      },
    },
  ],
  defaultSipHeaderNames: {
    contactCenterCallUuidHeader: "call-id",
    contactCenterCallUserInfoHeader: "x-customer-info",
  },
}
```

- **defaultSipHeaderNames:** Defines default SIP header names of various features integrated with contact center platforms. See customSipHeaderNames: Defines custom SIP headers that, if included in SIP messages, have a particular significance. below
- **Integrations:** The object structure is an array of contact center integration objects. Each object can include the following properties:
  - **contactCenterPlatform:** Defines the platform-type-name of this integration object. Possible values: "genesys_cloud", "genesys_pure_engage".
  - **serviceName:** Defines a unique name identifying the contact center integration service.
  - **enabled:** Defines whether this integration is available in the client or not. Default: false.
  - **customSipHeaderNames:** Defines custom SIP headers that, if included in SIP messages, have a particular significance.
    - **contactCenterCallUuidHeader:** Defines the SIP header name to be used, to denote the call UUID for that platform integration.
    - **contactCenterCallUserInfoHeader:** Defines the SIP header name to be used, to contain a customer-info JSON value.
- **Other platform-specific attributes:** For example, clientId / redirectUri / environment that are specific to allow Genesys Cloud sign-in. These require proper knowledge of the interoperability of a web client application with the contact center platform

Configure Authentication Providers for User Sign-In

The advancedOptions.js file allows you to set the authProvidersConfig object, which configures parameters of all the available authentication providers for the user to sign into. These authentication providers typically feature OpenID Connect / OAuth 2.0 sign-in, to allow the web client to obtain SIP credentials and other protected resources for the client functionality.

For example:

```
var authProvidersConfig =
{
  mandatoryProviderLoginName: "Genesys Cloud US-West-2",
  providers: [
    {
      providerLoginName: "Azure Active Directory",
      providerType: "AAD",
      authority: "https://login.microsoftonline.com/1911c65c-89....
      endpointUrl: "https://graph.microsoft.com/v1.0",
      realm: "",
      clientId: "e8a69733-d978-47b9-a938-cc8b260636a4",
      redirectUri: undefined,
      knownAuthorities: [],
      requestScopes: [],
      authRefreshRetriesForSipError: 5,
      forceLoginPromptOnEmptyCache: false,
      userInfoUrlRelativePath: "/me",
      tokenRevocationRelativePath: "",
      cacheLocation: "",
      sipAttributes: {
        SIP_DN_Attribute_name: "voip_identity_username",
        SIP_Password_Attribute_name: "voip_identity_password",
      },
    },
    {
      providerLoginName: "Keycloak",
      providerType: "Keycloak",
      authority: "https://keycloak.webrtc.audiocodes.com:8443",
      realm: "demo",
      clientId: "WebRTCDemo",
    },
  ]
}
```

- **mandatoryProviderLoginName:** Defines the providerLoginName attribute of the login provider which is mandatory. That is, the client, upon launch, always verifies that the user is logged in to it. Otherwise, it prompts a login to that specific provider without allowing you to continue. This can also be the serviceName property of a contact center integration that also serves as an authentication provider, e.g., Genesys Cloud.
- **Providers:** Defines the authentication providers that the user is allowed to sign into from the client GUI. They are represented as an array of elements, each corresponds to a single authentication provider:
  - **providerLoginName:** Defines the name of the authentication provider, as displayed as an entry in the client Account Settings. This name must be unique.

- **providerType:** Denotes the identity provider type and the mode of operation. Possible values:
  - **AAD (default):** Defines an Azure Active Directory endpoint
  - **Keycloak:** Defines a specific for the Keycloak identity platform
  - **OIDC:** Defines a general purpose provider that is Open Id Connect-compliant
- **endpointUrl:** Defines the identity-provider API endpoint URL. This is used for API calls to access resources protected by the OAuth token, such as a user identity profile. Default: https://graph.microsoft.com/v1.0 for "AAD" protocol mode, and the value for "authority" for OIDC mode. (Optional)
- **Authority:** Defines the endpoint from which tokens are obtained. For the AAD providerType, it should be **https://login.microsoftonline.com/<tenant-id>**, and for multi-tenant applications the tenant-id can be "common".
- **Realm:** Defines the optional authentication realm value. Mandatory for the 'Keycloak' providerType.
- **clientId:** Defines the identifier of the client application, as registered with the identity provider.
- **knownAuthorities:** Defines a list of optional known authorities. Required for providerType "AAD" if the "authority" value is not a commonly trusted authority such as https://login.microsoftonline.com.
- **redirectUri:** Typically and if not specified, this would be the client's site URL. It must be listed in the app registration for the "Single-page application" platform configuration. It can be left undefined.
- **requestScopes:** Defines the custom request scopes for obtaining access tokens. "User.Read" is automatically added for "AAD" protocol mode, and "openid" is added for other modes.
- **authRefreshRetriesForSipError:** Defines the number of times the client attempts to refresh the access token after an authentication error response from SIP registration. Default: 5.
- **forceLoginPromptOnEmptyCache:** When no cached OAuth account exists, for example when using session-storage:
  - **false** (default): Allows re-use of active session cookies if available, to save the user the need to enter credentials again.
  - **true:** Forces redirection to login page even though an active session cookie exists, to allow the user to log in to a different account if possible.
- **userInfoUrlRelativePath:** (Optional) The suffix is added to the endpoint URL for requesting the logged-in user info.
  - Default: '/me' for the 'AAD' providerType, and 'openid-connect/userinfo' for other providerType values.

- tokenRevocationRelativePath: Optional, the suffix added to the authority URL for revoking the user session on sign-out. No default value.
    - If this value is set to a non-empty path, then the client attempts to use it to invoke token revocation, so that all other login sessions with the same token are invalidated.
- **cacheLocation:** (Optional) Sets the preferred browser storage location. Default is the value under "storageConfig". Possible values:
    - **localStorage:** Best user experience, less secure. This allows the login session to persist across multiple browser sessions.
    - **sessionStorage:** Medium user-experience, more secure. On a new browser session, the login session does not exist, unless secure cookies are enabled and forceLoginPromptOnEmptyCache is false.
    - **memoryStorage:** Most secure, not using browser storage. Login session persists based on secure cookies, if enabled.
- **sipAttributes:** Defines the attribute names of SIP credential attributes that are assigned to the user profile once logged-in.

> The following two attributes must be unique in the user profile and are searched recursively throughout all the user's attribute hierarchy.

- **SIP_DN_Attribute_name:** (Optional) Defines the name of the id-token claim or user profile attribute, which defines the directory number / SIP username.
- **SIP_Password_Attribute_name:** (Optional) Defines the name of the id-token claim or user profile attribute, which defines the SIP password. If not defined, then the authentication flow uses the "Bearer <oauth-id-token>" as the SIP authorization header.

Migrating Authentication Providers and Contact Center Configuration from Previous Versions

Contact Center Integration – Genesys Pure Engage

To keep full support of the Genesys Pure Engage integration similarl to previous versions, especially when embedding the client within the Genesys Workspace Desktop Edition application, the **contactCenterConfig** object must contain the Genesys Pure Engage entry, exactly as it is in the example under Configure Integrations with Contact Center Platforms.

Authentication Provider Configuration

Backward Compatibility: The client is backward compatible with the previous authentication configuration, which uses the deprecated **defaultOAuthConfig** and **supportedAuthenticationSchemes** objects. However, this support is not guaranteed to last. We highly recommend to migrate to the new configuration structure using the **authProvidersConfig** object, as described below.

Migrating to the new configuration

- **SupportedAuthenticationSchemes:**

This configuration is deprecated and should be removed altogether. The included authentication scheme values are to be incorporated as follows:

- **"user-password"**: To support the user-password SIP registration authentication scheme, make sure that the attribute value **authProvidersConfig.mandatoryProviderLoginName** is an empty string or not set - i.e., there must not be a mandatory provider.
- **"oauth"/ "user-password_from_oauth":** This value is determined according to the authentication provider being used at runtime and whether its configuration includes the **sipAttributes** properties.
    - If **supportedAuthenticationSchemes** includes only a single scheme which is not user-password, then the authentication provider represented by **defaultOAuthConfig** should be set to be the value of **authProvidersConfig.mandatoryProviderLoginName**. (See below)

- **defaultOAuthConfig:** This object represents a single authentication provider, where Version 2.6.0 and later supports multiple providers.
    - The object **authProvidersConfig** should be created and include the following properties:
        - **providers:** This is an array of a single element which represents the provider from **defaultOAuthConfig** as follows:
        - **providerType:** If **defaultOAuthConfig.mechanismType** is "keycloak", set the value to "Keycloak". Otherwise, set the value to be the value of **defaultOAuthConfig.protocolMode**.
        - **Authority:** Set to be **defaultOAuthConfig.msal.authority** for **msal mechanismType**, or **defaultOAuthConfig.keycloak.oAuthUrl** for **keycloak mechanismType**.
        - **realm:** Set to be **defaultOAuthConfig.keycloak.realm** if keycloak is used, otherwise it is empty.
        - **authRefreshRetriesForSipError:** Same as **defaultOAuthConfig.accessTokenRefreshRetries**.
        - **forceLoginPromptOnEmptyCache:** Same as **defaultOAuthConfig.forceLoginPromptOnEmptyCache**.
        - **sipAttributes:** Should contain the SIP_DN_Attribute_name and SIP_Password_ Attribute_name properties and values from **defaultOAuthConfig.msal**.

- **redirectUri:** (Optional) Recommended to be left empty so that the client uses its current location URL. Alternatively, this can be set to be the same as **defaultOAuthConfig.msal.redirectUri**.
- All other properties, e.g., clientId, endpointUrl, etc., can either be set similarly as in **defaultOAuthConfig**, or according to the guidelines under Configure Authentication Providers for User Sign-In on page 40.

Client Resource Customization – Images, Sounds and Text

The following describes the client resource customization.

Logo

Client logos may be modified to reflect the customer company logo. The images may be found in the “images” folder. Customers should use the currently provided images as reference:

- The svg logo pixel dimensions is 241 X 34, and the logo-small image is 54 X 32 pixels. For logo-small this is optional, the logo-small can be of any size that’s not larger than the svg logo in either width or height.*
- The svg file can be easily generated from a bitmap image of the size 241 X 34 using standard image conversion tools. The svg in this case, should contain the viewBox attribute <0 0 241 43>, which is auto-generated during the creation of the image.*

Sound Files

- Client sound files for call-related rings, such as incoming call / outgoing call progress / call ended etc., can be modified by overriding the corresponding files under *webrtc_ client/sounds*.

Virtual Background Images

- The client includes several pre-packaged image files, that serve as default available virtual background images. These images are presented to the user in the Virtual Background settings, so they can select one of them to apply as a virtual background image effect.
- These images are contained in “images/virtual-backgrounds/”, as image01.jpg, image02.jpg and so on.
- These images can be modified by the customer if they wish to use other default images as available virtual backgrounds.

- The images must be replaced using the same filenames.
- The client does not support adding more images than currently exist.

Language and Text Customization

The client GUI text can be modified or translated to a different language. To do so, the client uses a text resource JSON file, which by default is located at *webrtc_ client/static/localization/strings-default.json*.

Text JSON Format and Behavior

- The text file format is a JSON object containing key-value entries of the following type: "KEY_NAME": "Corresponding text value".
- Each key can be assigned a different text value, which modifies the client text on screen.
- Each key with an empty value will prompt the client to use its own English default value.
- The client also has default English textual data, in case the text resource file is not in use or missing.

> The text customization file only allows you to replace existing text data with modified values. It does NOT cause any GUI orientation or layout changes according to the language.

Setting Client Lozalization

In the text JSON file, the "LANGUAGE_RESOURCE_LOCALE" entry denotes the localization that this text resource represents, which defaults to "en-US". This localization information notifies the client how to format language and region-related text data, such as date and time. The value to this key must be a valid IETF BCP 47 language-tag.

Setting a Different Language Resource

The client can be configured to fetch its text resource file from a different location URL. In order to do so, modify the **defaultLanguageResrouceURL in advancedOptions.js** property.

Resource Fetch Behavior

Each time the client page loads, it performs the following:

- Load custom string resource from cache, if exists
- Fetch custom string resource from defaultLanguageResrouceURL or from the default location
- Override the previous cache with the fetched custom string resource

Configure GUI Behavior with Regards to Text Resource Fetch

To prevent the client from showing its default English text GUI when waiting to download the text resource, the client can be configured to hide its GUI until it successfully loads a language resource from cache or by fetching it from the network.

Use the **waitForDefaultResourceFetchconfiguration** property in **advancedOptions.js** to determine whether the client GUI should be hidden until it has a language resource.

Browser Auto-Play Control for Tone Playback

Most browsers block media auto-play by default. This means that without user interaction on the page, no tone playback will be available, e.g., ringtones for incoming calls.

To support scenarios in which the client can play tones without user interactions, e.g., when the client page was just reloaded by the user and an incoming call arrives, the browser must allow auto-play for the client's website.

To enable auto-play for the browsers we officially support, you can either configure the browser settings or alternatively, configure the browser enterprise policies on Windows:

➢ **Google Chrome:**

- Browser Settings: Navigate to chrome://settings/content/sound, and add the web client URL to the "Allowed To Play Sound" list.
- Enterprise Policy Configuration: See https://chromeenterprise.google/policies/#AutoplayAllowlist

➢ **Microsoft Edge:**

- Browser Settings: Navigate to edge://settings/content/mediaAutoplay, and then add the web client URL to the "Allow" list.
- Enterprise Policy Configuration: See https://docs.microsoft.com/en-us/deployedge/microsoft-edge-policies#autoplayallowlist

➢ **Firefox:**

- Browser Settings: Navigate to the web client page, press Ctrl + I to show the page information, go to the Permissions tab and modify the "Auto-play" settings for that site to "Allow Audio and Video".

➢ **Safari:**

- Browser Settings: Navigate to the web client page. Go to the Safari menu, and then click **Settings for This Website**. In the displayed panel, under "Auto-Play", select Allow All Auto-Play.

Maintaining Web Client Activity When Browser Tab is in Background

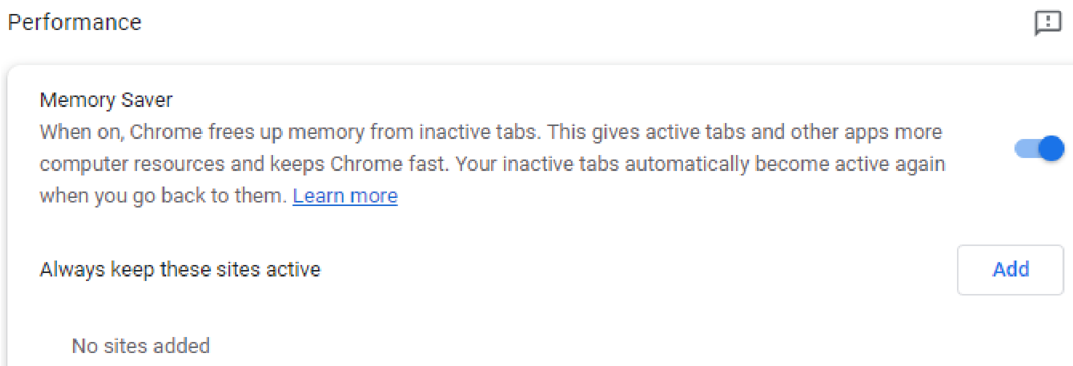
Modern browsers introduce new performance features that save resources, by automatically discarding tabs or putting them to sleep after being hidden for a certain period of time. Specifically, the introduction of new memory-saving features in both Chrome Memory Saver and Edge Sleeping Tabs, can negatively impact the client's ability to maintain active connections and receive incoming calls.

The client attempts to mitigate this by implementing approaches to keep the tab active. However, such endeavors are not reliable enough, because the browsers can still decide to pro-actively discard the tab without notifying the client or the user.

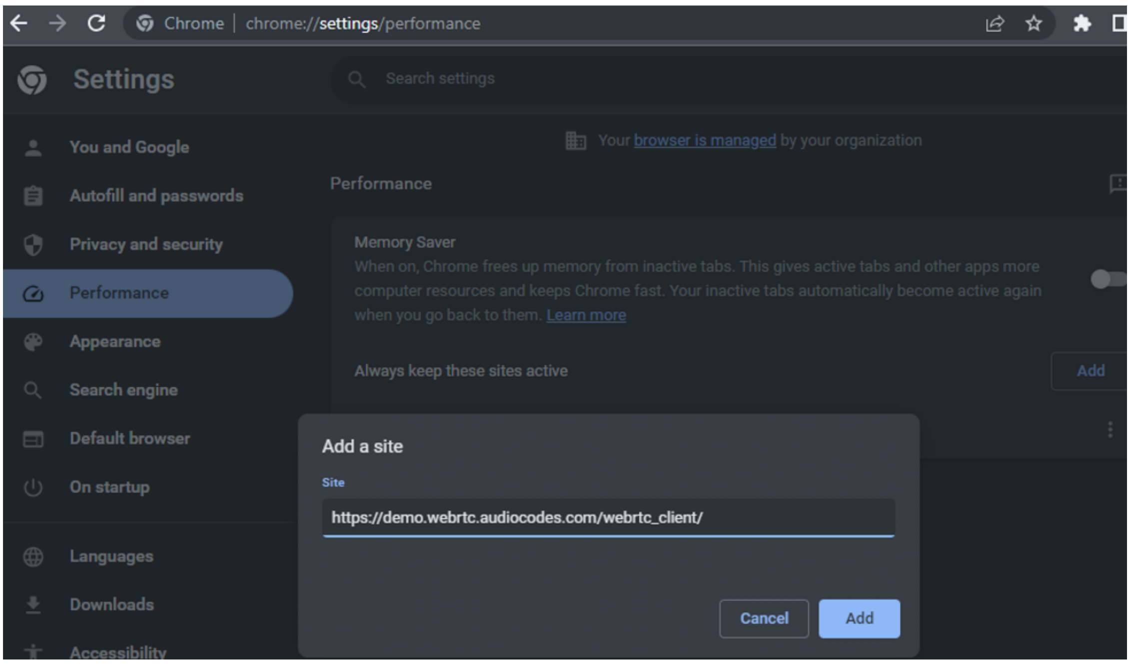
Currently, the best way to keep the tab active when hidden and avoid it from going to sleep or being discarded, is to disable memory-saving features for the client website as follows:

➢ To disable memory-saving features in Google Chrome:

1. Navigate to chrome://settings/performance; the following appears:

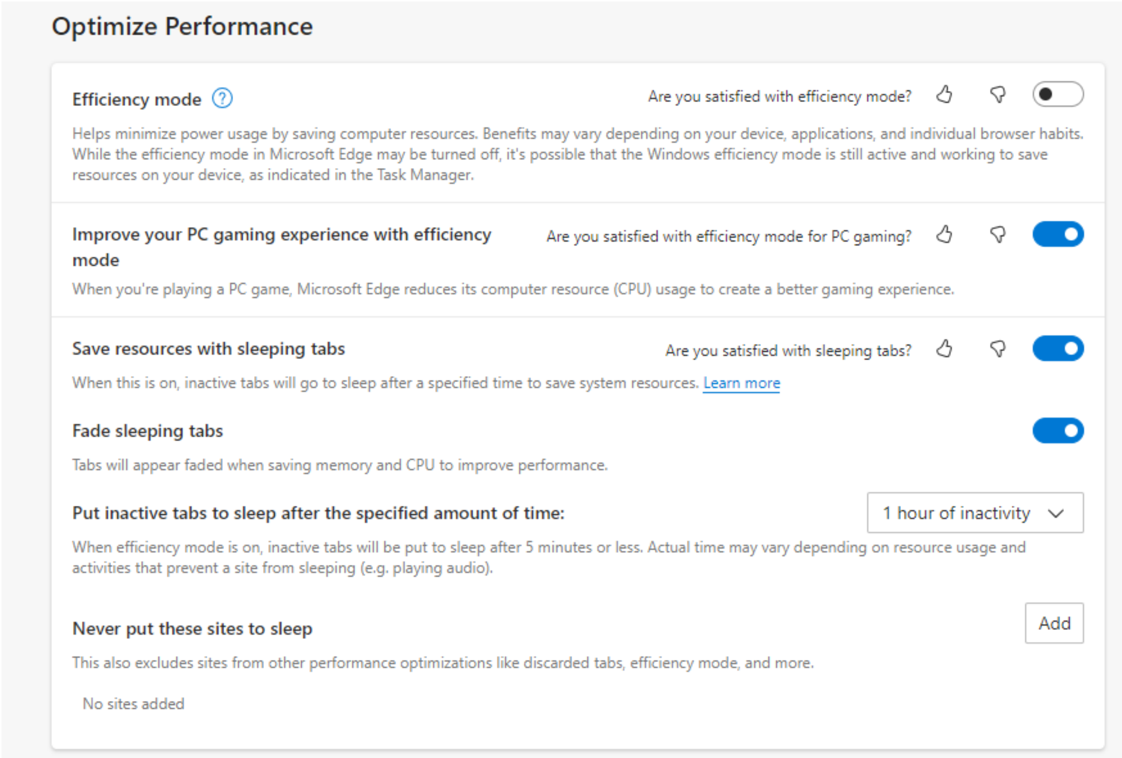


2. In the 'Always keep theses sites active' field, click **Add**.
3. In the dialog that opens, type the WebRTC web client full URL (including the schema), and then click **Add**.

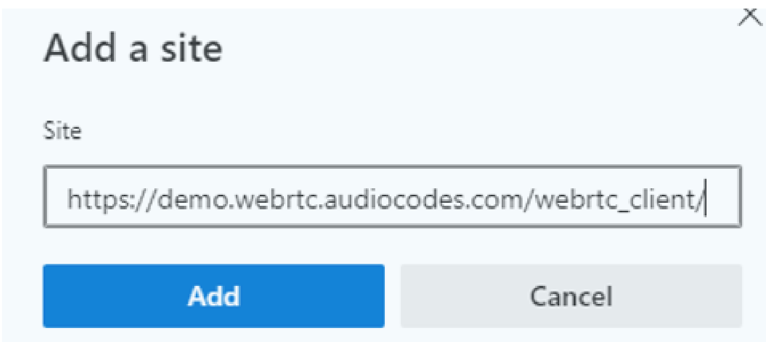


➢ To disable memory-saving features in Microsoft Edge:

1. Navigate to edge://settings/system.
2. In the 'Never put these sites to sleep' field, click **Add**.



3. In the dialog box, in the 'Site' field, enter the WebRTC web client full URL (including the schema), and then click **Add.**



NGINX and Apache Server Web Access Examples

Based on the configuration shown in this document, the WebRTC client can be accessed through a Web browser using the following URL examples:

- NGINX server: https://webrtcdemo.audiocodes.com/webrtc_client
- Apache server: https://ik.l5.ca/webrtc_client

The URLs above are AudioCodes demo sites and may not always be accessible.

Upgrading the Web Client to a New Release

When performing an upgrade, do the following to ensure that current modifications to configurations and resources are not lost.

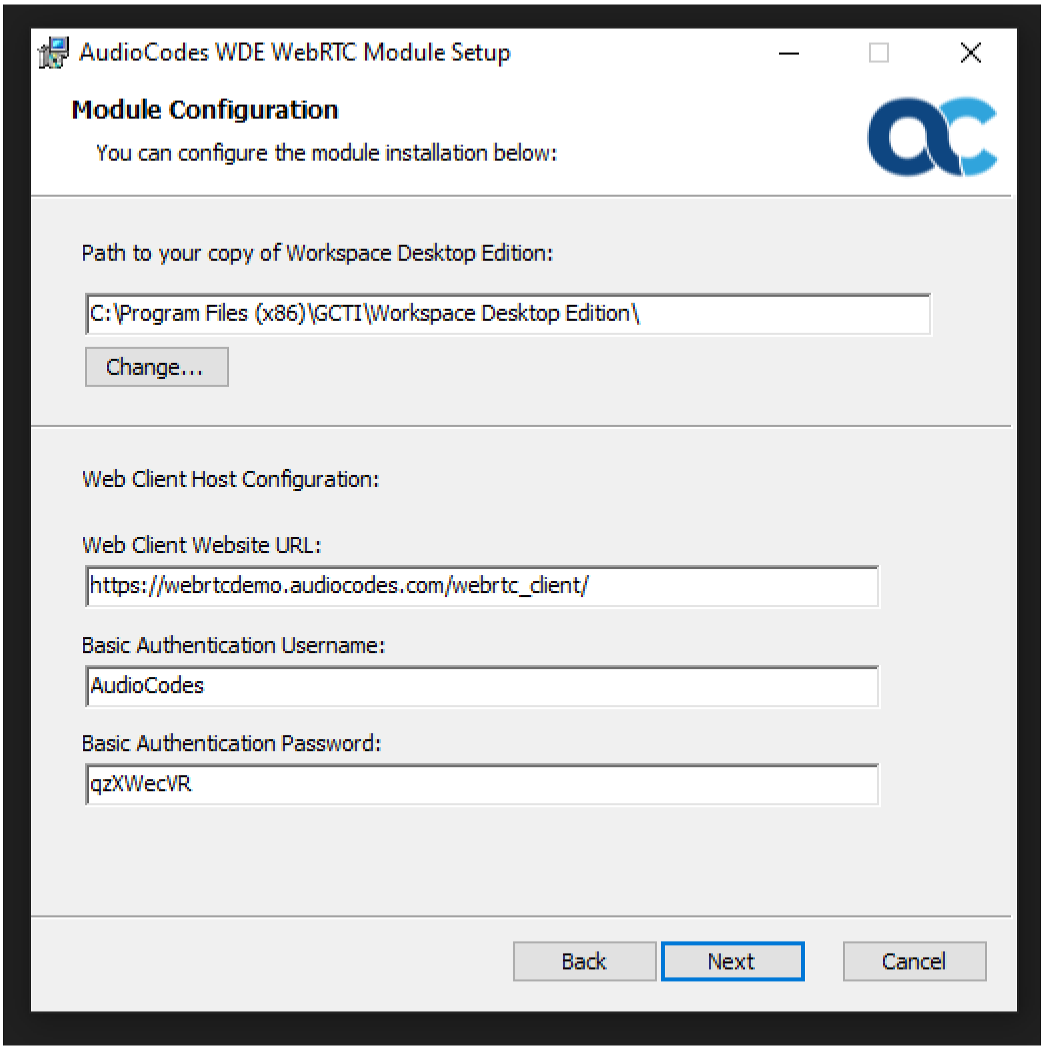
1. Maintain a list of all changes made to resources or configurations in the current release, that you wish to preserve when upgrading.
2. Backup the contents of the *webrtc_client* folder into *webrtc_client_old*.
3. Delete the entire contents of *webrtc_client*, and then copy into it the contents of the new build folder.
4. Preserve modified resources from the previous build by doing the following:
   a. Copy all files in *webrtc_client_old/images* that are listed in Step #1, and then paste and override them into *webrtc_client/images*.
   b. Copy all files in *webrtc_client_old/sounds* that are listed in Step #1, and then paste and override them into *webrtc_client/sounds*.
   c. Edit the file at *webrtc_client/static/localization/strings-default.json*, and then override all entries listed in Step #1 with their corresponding values from *webrtc_client_old/static/localization/strings-default.json*.
5. Preserve the client configuration by doing the following:
   a. For every configuration property in *webrtc_client/statis/js/advancedOptions.js*, that is listed in step #1, perform the following:
      i. If the configuration property is not marked as @deprecated, then modify its value to be the same as *webrtc_client_old/statis/js/advancedOptions.js.*
      ii. If the configuration property is marked as @deprecated, then use the documented recommendation next to it, to assign the modification to the alternative corresponding property.
6. Delete the *webrtc_client_old* folder.

4 Deploying AudioCodes WDE Extension

AudioCodes provides the WebRTC client as a WDE extension which is embedded in in the Genesys WDE. This client extension allows the WDE user (agent) to handle all the agent tasks in one application including the calls.

➢ To deploy the WDE extension:

1. Locate the WDE installation folder, e.g., C:\Program Files (x86)\GCTI\Workspace Desktop Edition\.
2. Unzip the file "ac_webrtc_wde.zip" to a temporary folder.
3. Run the installation wizard AudioCodes-WDE-WebRTC-Module-Setup.msi, and then follow the installation instructions (see figure below).
   a. To produce the installation log, use the command line to run the installer and configure the log file output, for example: msiexec /i AudioCodes-WDE-WebRTC-Module-Setup.msi /l*v myLog.txt.
   b. Web Client Host Configuration:
      - Web Client Website URL (mandatory): The URL of the WebRTC client.
      - Basic Authentication Username (optional): If the HTTP requires basic authentication to enter the page this value is for username.
      - Basic Authentication Password (optional): If the HTTP requires basic authentication to enter the page this value is for password.


AudioCodes WDE WebRTC Module Setup
Module Configuration
You can configure the module installation below:
Path to your copy of Workspace Desktop Edition:
C:\Program Files (x86)\GCTI\Workspace Desktop Edition\
Change...
Web Client Host Configuration:
Web Client Website URL:
https://webrtcdemo.audiocodes.com/webrtc_client/
Basic Authentication Username:
AudioCodes
Basic Authentication Password:
qzXWecVR
Back
Next
Cancel

5 Configuring Genesys WebRTC DN Object Endpoint

You can configure Genesys WebRTC DN (Directory Number) objects using various options, as shown in the following example:

```
contact = *
dual-dialog-enabled = false
enable-agentlogin-presence = false
enable-agentlogin-subscribe = true
make-call-rfc3725-flow = 1
refer-enabled = false
rfc-2976-dtmf = true
session-refresh-interval = 90
sip-cti-control = talk,hold,dtmf
transfer-complete-by-refer = false
use-register-for-service-state = true
```

The following table describes the DN objects options:

Table 5-1: DN Objects Descriptions

Option	Values	Description	Comment
authenticate-requests	register, invite	Specifies whether incoming SIP requests are treated with an authentication procedure under the following conditions: ■ The name of the incoming SIP message exists in the list of the authenticate-requests parameter. ■ The option password is configured on the same DN object.	Defined on the DN object, not on the agent ID object

Option	Values	Description	Comment
contact	No default SIP URI *	The contact URI that the SIP Server uses for communication with the endpoint.	If it is defined as "*", the SIP Server uses the contact URI it receives when considering self-registering a WebRTC endpoint.
dual-dialog-enabled	true (default) false	Provides the functionality to make consultation calls for endpoints that can only accept one active SIP dialog. Set the value to "false" for endpoints that accept only one active SIP dialog.	Must be set to "false" for the WebRTC client endpoint.
enable-agentlogin-subscribe	false (default) true	Enables SIP Server control over the state of an agent based on SIP messages that are received from the agent endpoint. The SIP server can log in or log out an agent, in response to SIP SUBSCRIBE requests. It can also change the availability state for an agent in response to NOTIFY requests. To enable this functionality, set this option to "true". To disable the functionality, set the option to	This must be set to "false" when considering the Genesys Business Continuity feature. Genesys Business Continuity is not supported in the initial release of the Genesys WebRTC client

Option	Values	Description	Comment
		"false".	
make-call-rfc3725-flow	1, 2	Controls which SIP call flow to choose when a call is initiated by a RequestMakeCall. The specified value is equal to the Call Flow number as described in RFC 3725. Only Flow 1 and Flow 2 from RFC 3725 are currently supported. Note: This option is enabled only when the option refer-enabled is set to "false" for that DN. See refer-enabled for more information on this option.	Both flow definitions work with WebRTC client endpoint.
Password	String	Specifies the password for SIP endpoint registration with the local registrar. If it is present, registration attempts are challenged, and the password is verified. If it is not present, the registration is not challenged. The realm for password authentication is configured globally; there is one realm per SIP Server.	Defined on the DN object, not on the agent ID object

Option	Values	Description	Comment
refer-enabled	▪ true (default) ▪ false	Set this option to "false" for the SIP Server to use a re-INVITE request method when contacting the softswitch.	This must be set to "false" for the WebRTC client endpoint.
rfc-2976-dtmf	▪ false (default) ▪ true	When this option is set to "true" in a particular DN configuration, the SIP server sends DTMF tones in RFC 2976 format, to that device using the INFO request method when an agent issues a TSendDTMF request.	For the WebRTCclient configured to 'true'
session-refresh-interval	▪ 1800 (default) ▪ 0, 90-86400	Specifies (in seconds) how often active calls are checked to see if they are still active. A 0 (zero) value disables this feature (the session refresh mechanism is turned off). Values between 1 and 89 (inclusive) are treated as value 90. This option is used to remove stuck calls that must accumulate, if endpoints terminate calls without sending the appropriate SIP	As needed. The default value is 1800.

Option	Values	Description	Comment
		message.	
sip-cti-control	■ No default ■ talk, hold, dtmf	Specifies the behavior of the DN which represents a SIP endpoint which supports the BroadSoft SIP Extension Event Package. When set to "talk", the TAnswerCall request is issued against the DN, which means that the call is answered remotely by a T-Library client. Otherwise, the TAnswerCall request is not supported. When set to "hold", the endpoint is put on hold by a NOTIFY hold message. Note: "talk, hold, dtmf" could be used simultaneously as a list of comma-separated values.	For 3pcc, depending on actions, consider setting "talk, hold, dtmf".
transfer-complete-by-refer	■ false (default) ■ true	If set to "true", this option enables the SIP server to complete a two-step transfer by sending a REFER message to the party in the primary call. The SIP server uses the same content as in the REFER message that is sent for a single-step transfer.	This must be set to "false" for the WebRTC client endpoint.

Option	Values	Description	Comment
		For this option to work, you must configure refer-enabled on the Trunk DN to "true".	

6 Configuring Citrix Remote Desktop for Integration with WebRTC Client

To enable Citrix remote desktop deployment, the remote Citrix machine must be configured as follows:

1. The following registry modifications must be applied:
   a. Enable Citrix redirection:
      - Key Path: HKCU\Software\Citrix\HDXMediaStream
      - Key Name: MSTeamsRedirSupport
      - Key Type: DWORD
      - Key Value: 1
   b. Add the Chrome program to the allow list:
      - Key Path: HKLM\Software\WOW6432Node\Citrix\WebSocketService
      - Key Name: ProcessWhitelist
      - Key Type: MULTISZ
      - Key Value: chrome.exe
   c. (Optional) Configure Citrix logging:
      - Key Path: Computer\HKEY_CURRENT_USER\Software\Citrix\HDXMediaStream
      - Key Name: WebrpcLogLevel
      - Key Type: DWORD
      - Key Value: 0

> The log files are created in the local machine, not at the remote Citrix machine.
> For each RTP session, a log directory is created with a timestamp.
> You can find log files for RTP sessions at the path:
> %temp%\HdxRTCEngine\<session-timestamp>\

2. Configure Microphone Privacy Settings: In Citrix Desktop Windows, open Microphone Privacy Settings, and then enable microphone usage.

7 Configuring SBC WebRTC

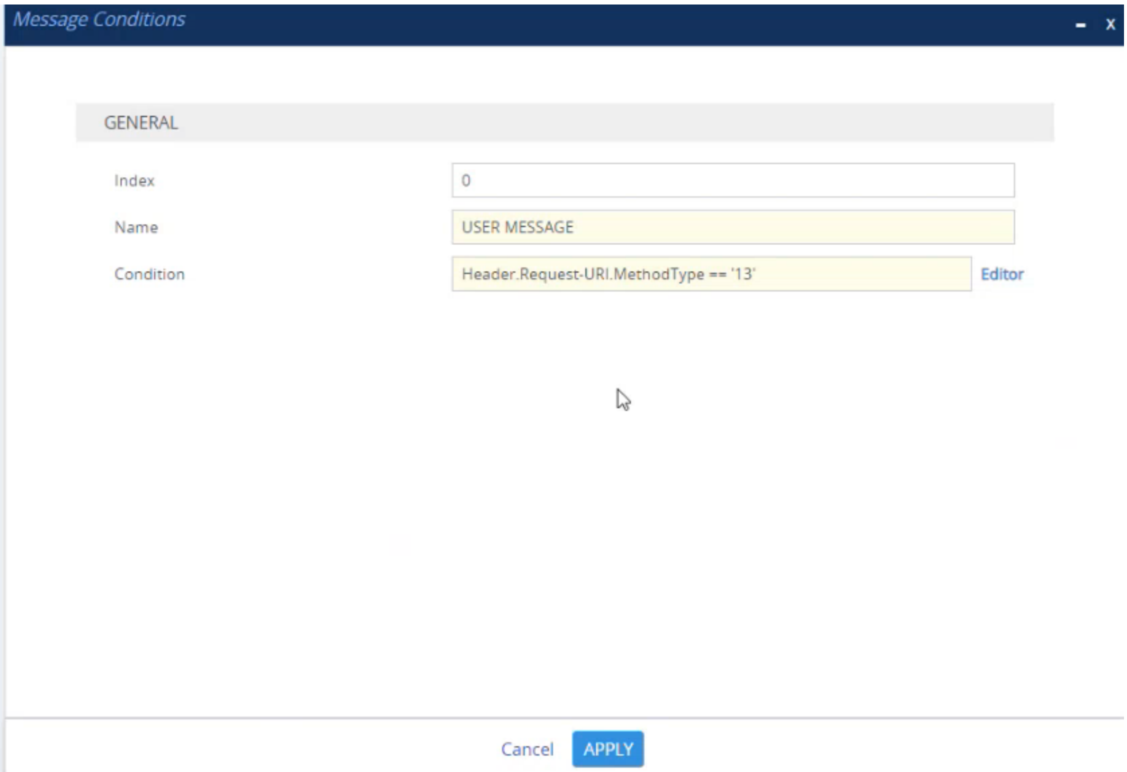
The procedures below describe how to configure SBC WebRTC.

Creating an IP-to-IP Message Route

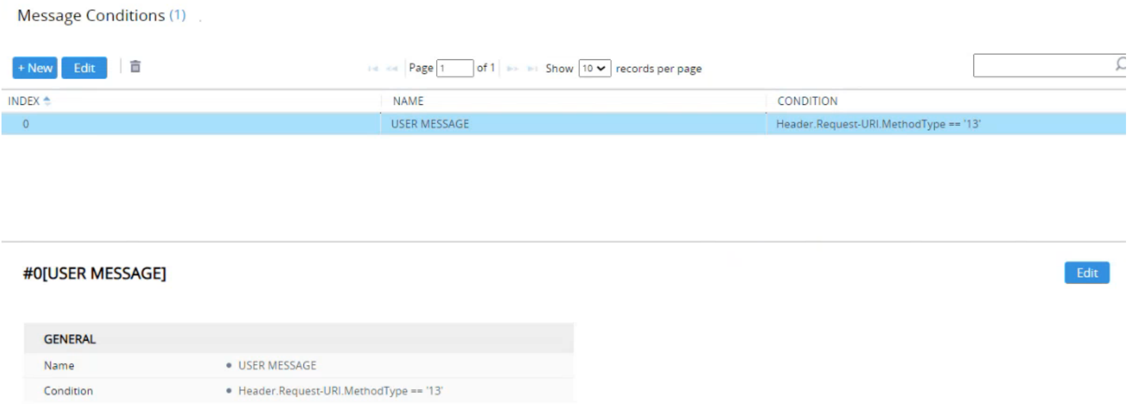
The following describes how to create an IP-to-IP message route.

➢ **To create an IP-to-IP Message Route:**

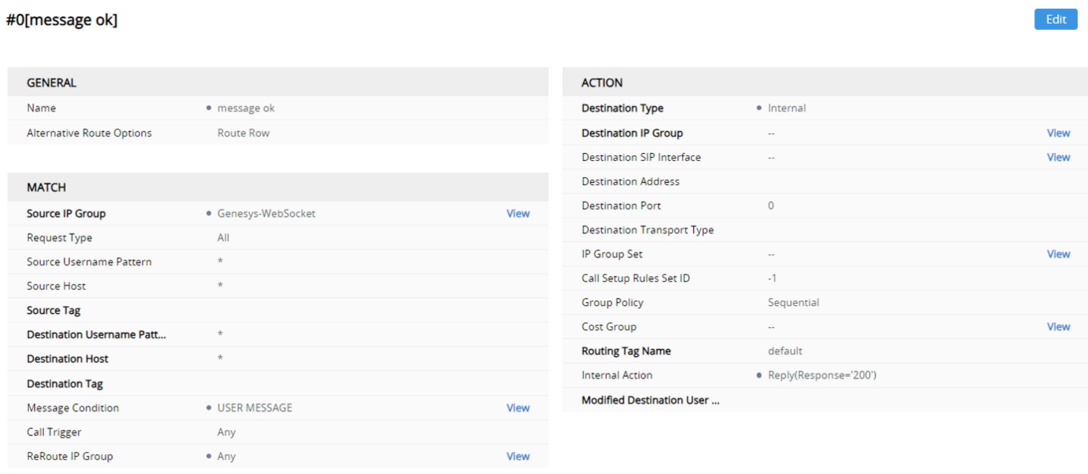
1. Open the Message Conditions table (**Setup** menu > **Message Manipulation** folder > **Message Conditions**).
2. Click **New**; the following appears:



3. In the 'Name' field, enter "USER MESSAGE".
4. In the 'Condition' field, enter the condition, and then click **APPLY**; the message condition has been created.



5. Open the IP-to-IP routing page (**Setup** menu > **SBC** folder > **Routing > IP-to-IP Routing**) and then click **New**; the following appears:



6. From the 'Message Condition' drop-down list, select **USER MESSAGE**, and then click **View**; the following appears:

7. In the 'Condition' field, enter the condition, and then click **Save**.

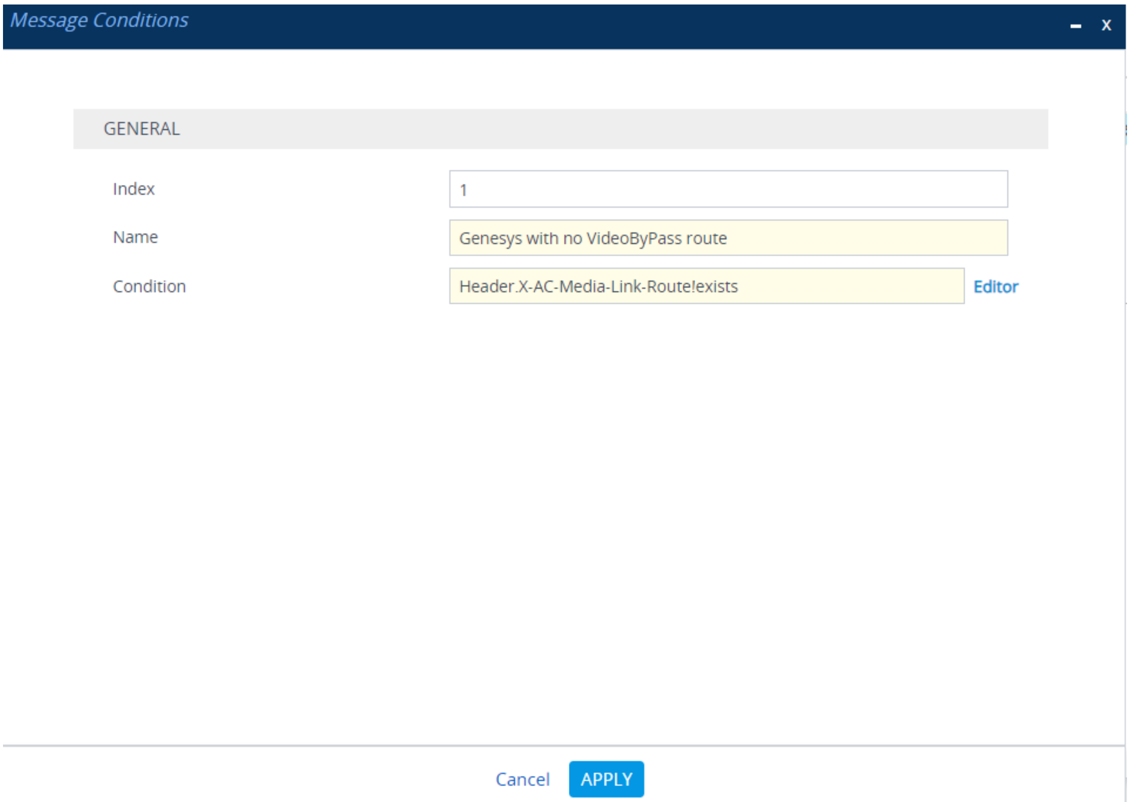
Creating a Call (Voice Only) Route

The following describes how to create a Call (Voice Only) Route.

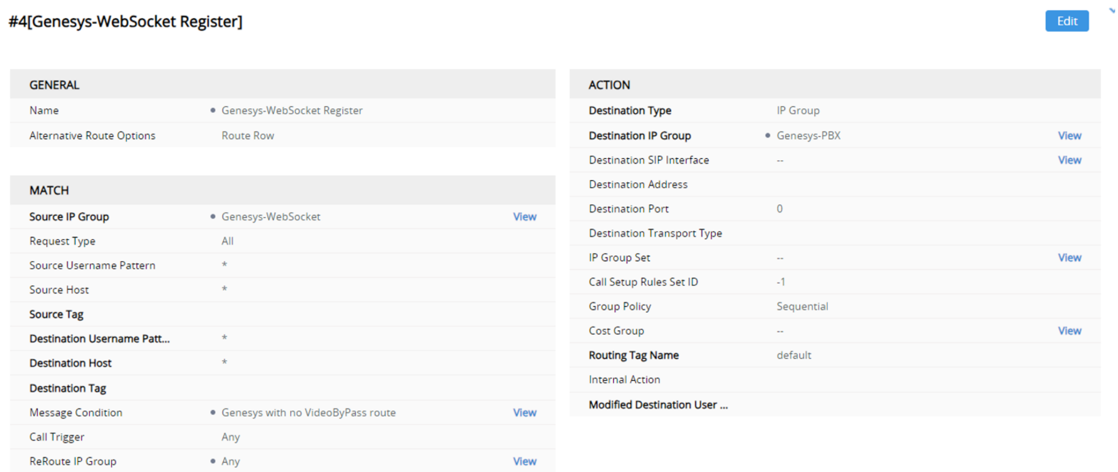
➢ **To create a Call route for voice calls with a Message Condition:**

1. Open the Message Conditions table (**Setup** menu > **Message Manipulation** folder > **Message Conditions**).

2. Click **New**; the following appears:



3. In the 'Name' field, enter "Genesys with no VideoByPass route".
4. In the 'Condition' field, enter the condition, and then click **APPLY**; the message condition has been created.
5. Open the IP-to-IP routing page (**Setup** menu > **SBC** folder > **Routing > IP-to-IP Routing**) and then click **New**; the following appears:



6. In the 'Name' field, enter "Genesys-WebSocket-Register".
7. In the 'Alternative Route Options' field, enter "Route Row".
8. From the 'Message Condition' drop-down list, select **Genesys with no VideoByPass route**.
9. From the 'Destination Group' drop-down list, select **IP Group**.

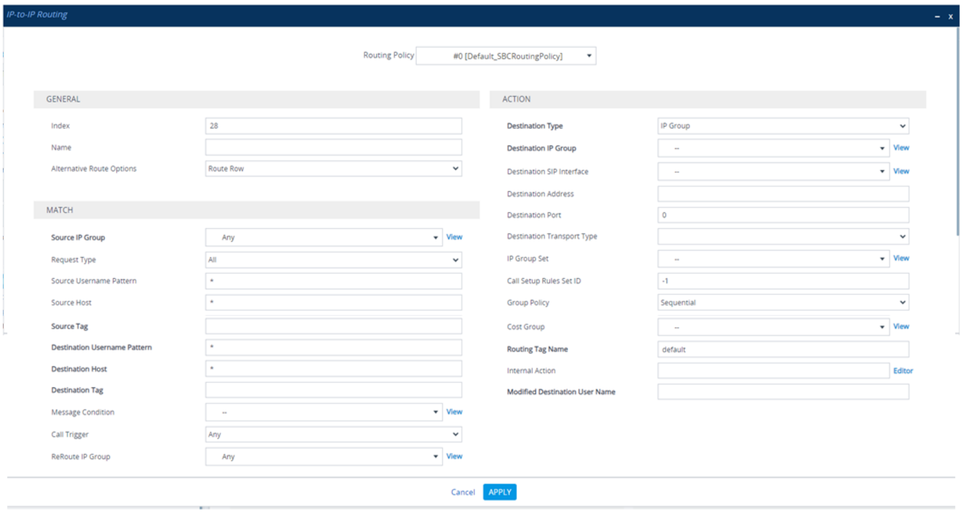
10. In the 'Destination IP Group' drop-down list, select **Genesys-PBX**.
11. Click **Apply**.

Creating an IP-to-IP Route for Video Calls

The following describes how to create an IP-to-IP Route for video calls.

➢ To create an IP-to-IP Route for Video Calls:

1. Open the IP-to-IP table (**Setup** menu > **SBC** folder > **Routing > IP-to-IP Routing**).
2. Click **New**; the following appears:



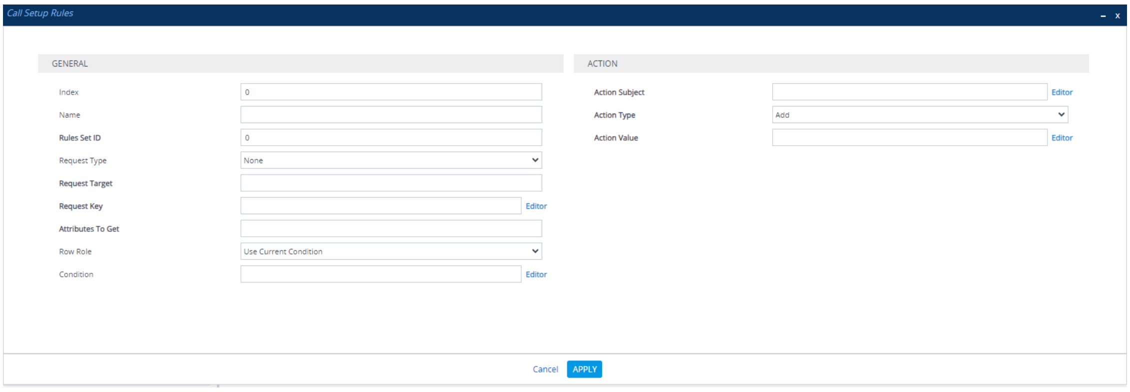
3. In the 'Name' field, enter "Genesys-Websocket with Video".
4. From the 'Alternative Route Options' drop-down list, select **Route Row**.
5. From the 'Source IP Group' drop-down list, select **Genesys-WebSocket**.
6. From the 'Destination Type' drop-down list, select **IP Group**.
7. From the 'Destination IP Group' drop-down list, select **Genesys-WebSocket**.
8. Click **Apply**.

Configuring Call Setup Rules

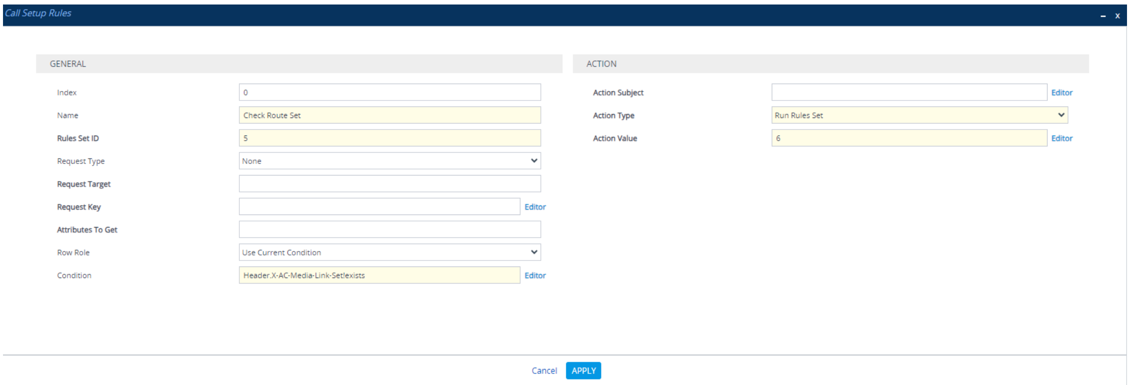
The following describes how to configure Call Setup rules.

➢ To configure Call Setup Rules:

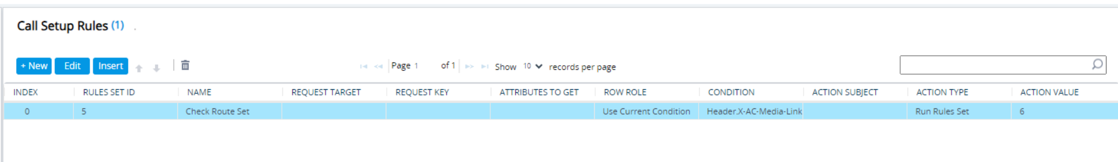
1. Open the Call Setup Rules table (**Setup** menu > **SIP Definitions** folder > **Call Setup Rules**).
2. Click **New**; the following appears:



3. Create Setup Rule #0 with the following:
   a. In the 'Name' field, enter "Check Route Set".
   b. In the 'Rules Set ID' field, enter "5".
   c. In the 'Row Role' field, enter "Use Current Condition".
   d. In the 'Condition' field, enter "Header.X-AC-Media-Link-Set!exists, and then click **Save**.
   e. In the 'Action Type' field, enter "Run Rules Set".
   f. In the 'Action Value' field, enter "6".
   g. Click **APPLY**.



The following appears:



4. Create Setup Rule #1 with the following.
   a. In the 'Name' field, enter "X-Header_DB_Set".
   b. In the 'Rules Set ID' field, enter "5".
   c. In the 'Request Type' field, enter "HTTP GET".
   d. In the 'Request Target' field, enter "X-Header_DB_SET".

   e. In the 'Request Key' field, enter " Header.X-AC-Media-Link-Set+'/'+Header.From.URL+'/ex/3600'".
   f. In the 'Row Role' field, enter "Use Current Condition".
   g. In the 'Action Subject' field, enter " Header.temp".
   h. In the 'Condition' field, enter "Header.X-AC-Media-Link-Set!exists".
   i. In the 'Action Type' field, enter "Add".
   j. In the 'Action Value' field, enter "HTTP.Response.Body".
   k. Click **APPLY**.
5. Create Setup Rule #2 with the following.
   a. In the 'Name' field, enter "check route".
   b. In the 'Rules Set ID' field, enter "5".
   c. In the 'Row Role' field, enter "Use Current Condition".
   d. In the 'Condition' field, enter " Header.X-AC-Media-Link-Route !exists".
   e. In the 'Action Type' field, enter "Add".
   f. In the 'Action Value' field, enter "HTTP.Response.Body".
   g. In the 'Action Type' field, enter "Exit".
   h. In the 'Action Value' field, enter "true".
   i. Click **APPLY**.
6. Create Setup Rule #3 with the following.
   a. In the 'Name' field, enter "X-AC-Media-Link-Route".
   b. In the 'Rules Set ID' field, enter "6".
   c. In the 'Request Type' field, enter "HTTP GET".
   d. In the 'Request Target' field, enter "X-Header_DB_GET".
   e. In the 'Request Key' field, enter "Header.X-AC-Media-Link-Route".
   f. In the 'Row Role' field, enter "Use Current Condition".
   g. In the 'Condition' field, enter "HTTP.Response.Body regex (.*)(sip:)(.*)(.@)(.*)(").*".
   h. In the 'Action Subject' field, enter "param.call.dst.User".
   i. In the 'Action Type' field, enter "Modify".
   j. In the 'Action Value' field, enter "$3".
   k. Click **APPLY**.
7. Create Setup Rule #4 with the following.
   a. In the 'Name' field, enter "X-AC-Media-Link-Route".

b. In the 'Rules Set ID' field, enter "6".

c. In the 'Row Role' field, enter "Use Previous Condition".

d. In the 'Action Subject' field, enter "param.call.dst.host".

e. In the 'Action Type' field, enter "Modify".

f. In the 'Action Value' field, enter "$5".

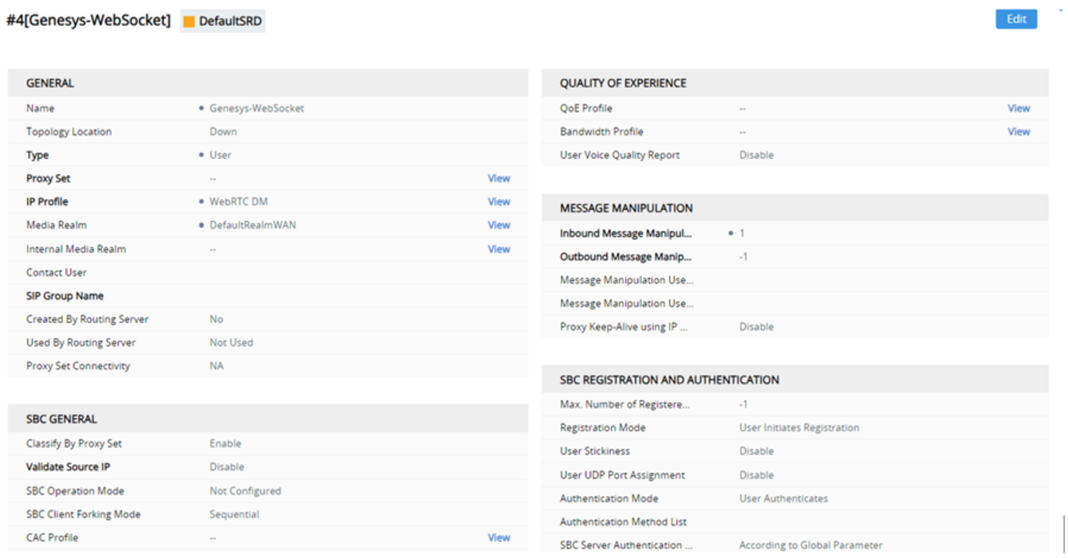
g. Click **APPLY**.

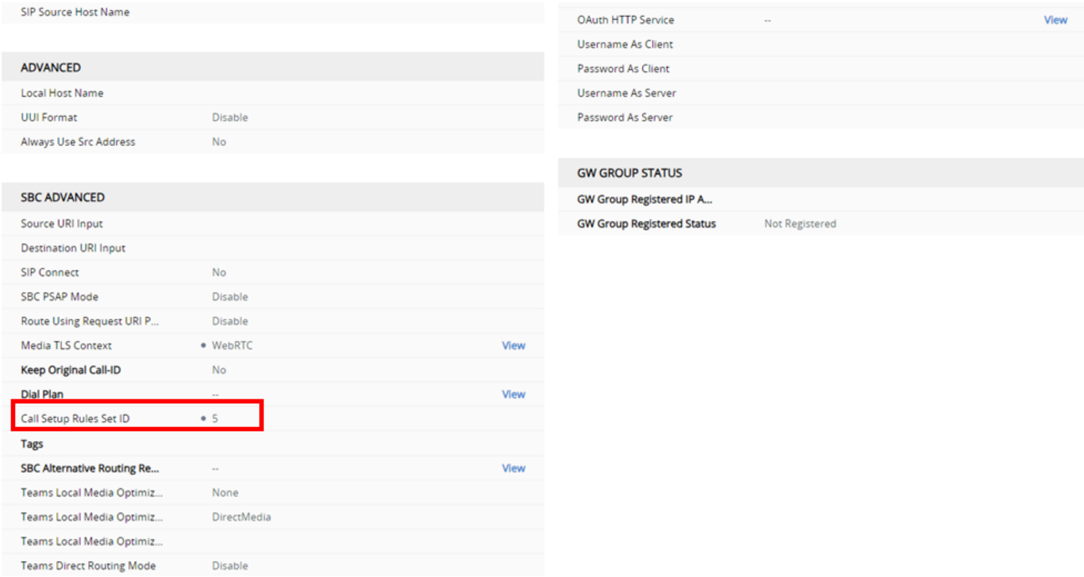
Configuring IP Groups

The following describes how to configure IP groups.

➢ **To configure IP Groups:**

1. Open the IP Groups page for **Genesys-WebSocket** (**Setup** menu > **Core Entities** folder > **IP Groups**).
2. In the 'Call Setup Rules ID', enter "5", and then click **APPLY**.




SIP Source Host Name
ADVANCED
Local Host Name
UUI Format Disable
Always Use Src Address No
SBC ADVANCED
Source URI Input
Destination URI Input
SIP Connect No
SBC PSAP Mode Disable
Route Using Request URI P... Disable
Media TLS Context • WebRTC View
Keep Original Call-ID No
Dial Plan -- View
Call Setup Rules Set ID • 5
Tags
SBC Alternative Routing Re... -- View
Teams Local Media Optimiz... None
Teams Local Media Optimiz... DirectMedia
Teams Local Media Optimiz...
Teams Direct Routing Mode Disable
OAuth HTTP Service -- View
Username As Client
Password As Client
Username As Server
Password As Server
GW GROUP STATUS
GW Group Registered IP A...
GW Group Registered Status Not Registered

8 Troubleshooting Client Connections

The following sections describes troubleshooting client connections.

Client Communication with the SBC

- The first step for troubleshooting SBC connections or registration issues, is to prevent the browser from automatically freezing or discarding the web client to save resources. See Maintaining Web Client Activity When Browser Tab is in Background on page 46.
- It's highly recommended to use a certificate issued by a public CA for the SBC.
- If the certificate is provided internally, the following pre-requisites must be fulfilled:
  - The certificate must contain a FQDN (may also be resolved by internal DNS name) for the SBC
  - FQDN in the Certificate must include SAN (Subject Alternative Name)
  - The browser's trusted root certificates store must contain a certificate in the same trusted certificate chain, as the internally generated certificate

Client HTTP Secure Connectivity

The WebRTC web client may connect to various services via HTTPS. Typically, the browser might reject these service connections if they do not meet proper security requirements.

To resolve HTTPS certificate issues:

- Use a valid certificate issued by a public CA.

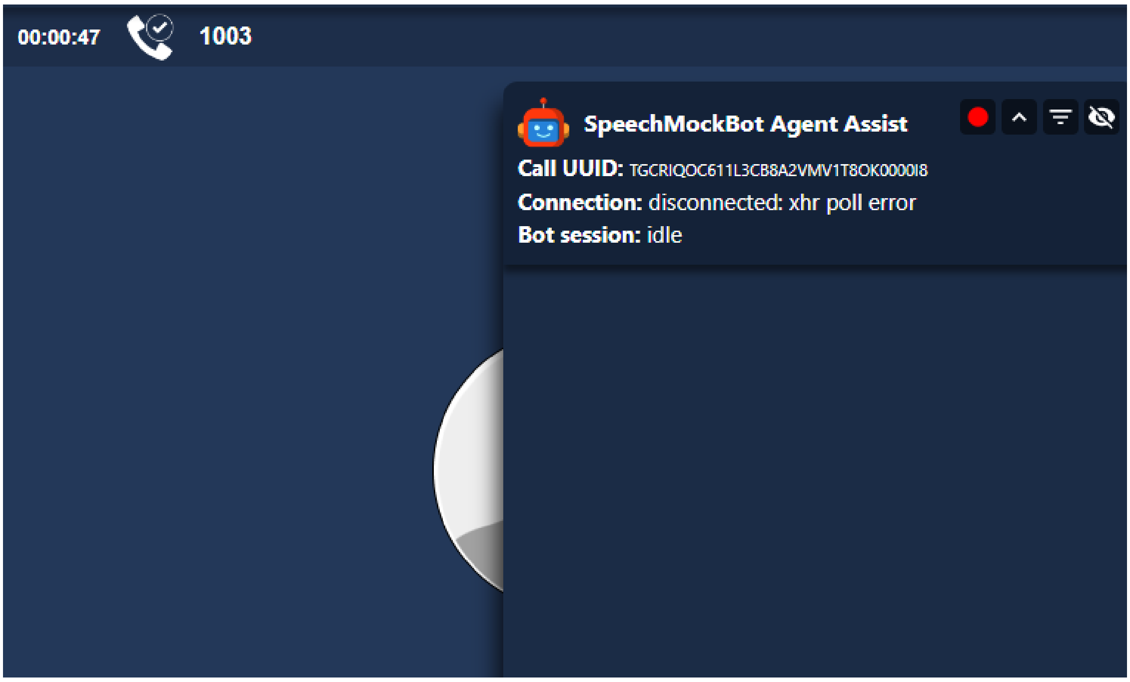
  or

- For self-signed certificate, or one issued by the organization’s internal CA:

  The browser might deem the certificate as invalid, in which case it must be configured to accept it for the specific HTTPS request domain. See below for a detailed example.

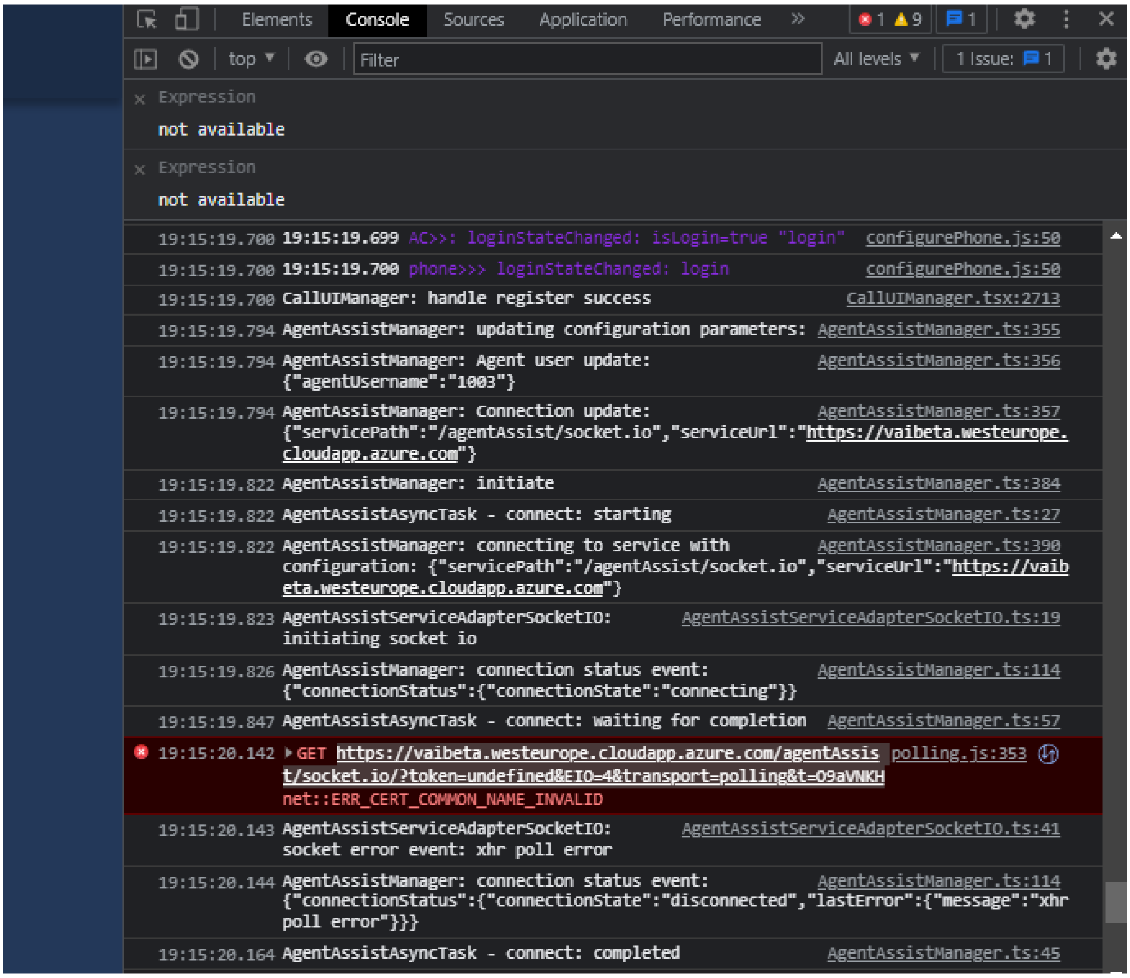
➢ **Example:**

1. The client might show the following connection error when the Agent-Assist service is being used:

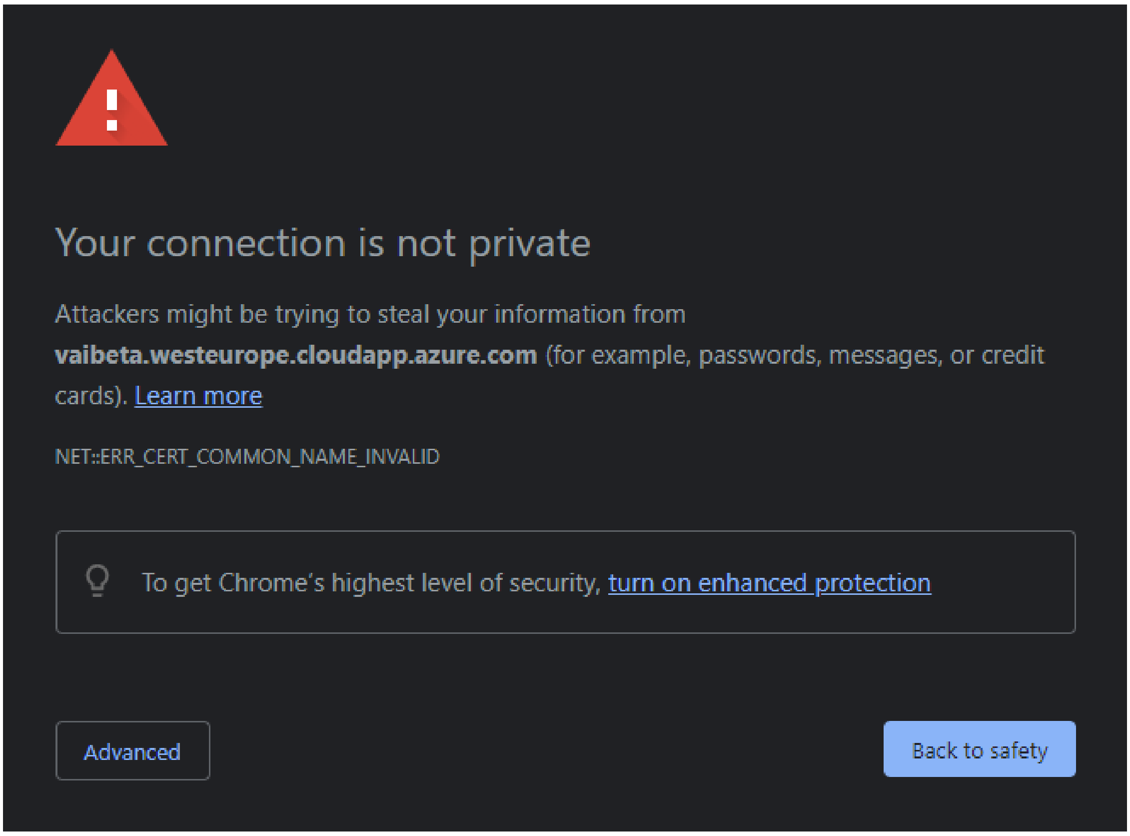


2. To diagnose and verify the underlying error, open the browser console by opening the web-inspector (right-click and then choose **inspect**).

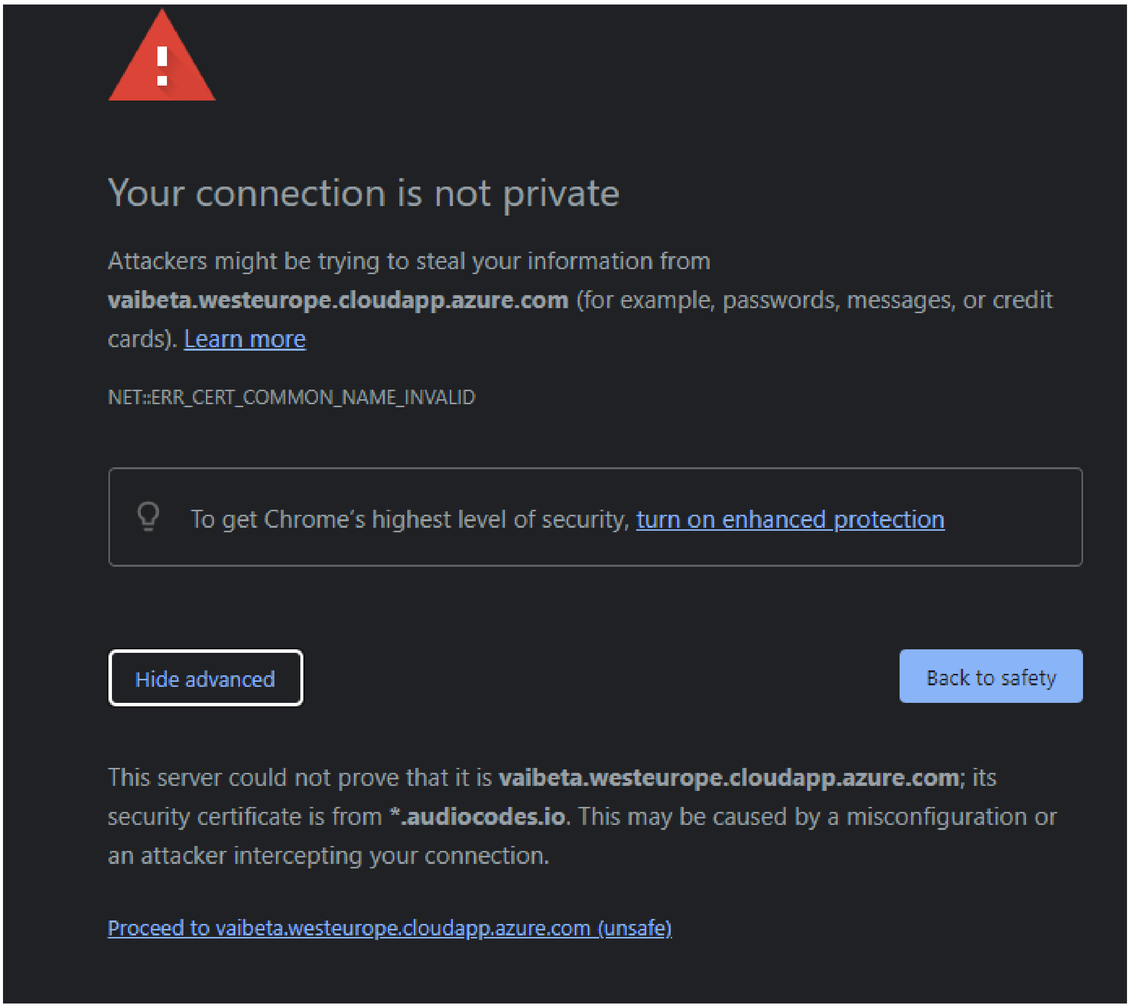
3. Select the “console” tab; an error message such as the following might appear:



4. To accept this certificate, copy the request URL to a new tab and navigate there. The following message will appear:



5. Click the **Advanced** button to show the following:



6. Click "Proceed to ……. "; the browser accepts the certificate.
7. Reloading the client uses the applied configuration, and the connection will succeed.

This page is intentionally left blank.

International Headquarters

1 Hayarden Street,

Airport City

Lod 7019900, Israel

Tel: +972-3-976-4000

Fax: +972-3-976-4040

AudioCodes Inc.

80 Kingsbridge Rd

Piscataway, NJ 08854, USA

Tel: +1-732-469-0880

Fax: +1-732-469-2298

Contact us: https://www.audiocodes.com/corporate/offices-worldwide

Website: https://www.audiocodes.com/

Documentation Feedback: https://online.audiocodes.com/documentation-feedback

Document #: LTRT-14163