



## VXML Interaction Server Maintenance Guide Version 5.0.3

This page was not added to the PDF due to the following tag(s): article:topic-guide

## Configuring HTTPS for VXML Interaction Server and Platform ToolKit

### Overview

HTTP is the default protocol for defining how messages are formatted, transmitted and what actions should take place between web browsers and web servers. HTTPS is essentially the HTTP protocol used within the SSL/TLS protocol, which provides a secure connection between two machines operating over the Internet or an internal network. Enabling the VXML Interaction Server (VIS) to communicate with Platform Toolkit (PTK) over HTTPS requires three steps:

1. Configure HTTPS between the voice browser and VXML Interaction Server.
2. Configure HTTPS between the Web browser and Platform Toolkit.
3. Configure HTTPS between the Java Web server and Platform Toolkit.

### Configure HTTPS Between the Voice Browser and VXML Interaction Server

To configure HTTPS between the voice browser and VXML Interaction Server:

1. From a command prompt, enter the following command:  
**"C:\Program Files (x86)\Java\jre6\bin\keytool" -genkey -alias tomcat-keyalg RSA -dname "cn=Virtual Hold Technology, ou=Product Development, o=VHT, c=US" -keystore "C:\Users\Developer\keystore"**  
where:  
*C:\Program Files (x86)\Java\jre6\bin* = Location of Java used by Apache Tomcat  
*tomcat* = Friendly name of created certificate  
*Virtual Hold Technology* = Company name  
*Product Development* = Organizational unit  
*VHT* = Organization name  
*US* = Country name  
*C:\Users\Developer\keystore* = Path to the created keystore
2. Enter the desired password for the keystore when prompted.
3. Enter the desired password for the created certificate when prompted.
4. Open the server.xml file found in the Tomcat installation directory. The default file location is C:\Program Files (x86)\Apache Software Foundation\Tomcat 7.0\conf\server.xml.
5. Uncomment the Connection Port 8443 section.

```
<!--  
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  
          maxThreads="150" scheme="https" secure="true"  
          clientAuth="false" sslProtocol="TLS" />  
-->
```

6. Add the `keystoreFile`, `keystorePass` and `keyAlias` attributes by entering the following line to the Connector tag in the Connector Port 8443 section:

```
keystoreFile="C:/Users/Developer/.keystore" keystorePass="changeit" keyAlias="tomcat"
```

where:

`C:/Users/Developer/.keystore` = Path to the created keystore (refer to Step 1)

`changeit` = Password for the keystore (refer to Step 2)

`tomcat` = Friendly name of created certificate (refer to Step 1)

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  
    maxThreads="150" scheme="https" secure="true"  
    keystoreFile="C:/Users/Developer/.keystore" keystorePass="changeit" keyAlias="tomcat"  
    clientAuth="false" sslProtocol="TLS" />
```

7. Save the `server.xml` file.
8. Open the `toolkit.properties` file in a text editor. The default file location is `C:\Virtual Hold`.
9. In the `#Name File Configuration` section, change the `webaudio` path to `//10.10.1.61.8443/`.
10. Save and close the `toolkit.properties` file.
11. Restart Tomcat.

## Configure HTTPS Between the Web Browser and Platform Toolkit

To configure HTTPS between the web browser and Platform Toolkit:

1. Open the Internet Information Services (IIS) Manager.
2. In the Connections pane, select the Web server.
3. Obtain a software certificate. If necessary, perform the following to create a self signed certificate:
  - a. In Features view, double-click **Server Certificates**.
  - b. In the Actions pane, click **Create Self-Signed Certificate**.
  - c. On the Create Self-Signed Certificate page, enter a name for the certificate in the Specify Friendly Name window (refer to Step 1 of the previous section).
  - d. Click **OK**.
4. In the Connections area, select the Web site used by the Platform Toolkit.
5. In the Actions pane, click **Bindings**.
6. In the Site Bindings dialog box, click **Add**.
7. In the Add Site Binding dialog box, in the Type field, select **https**.
8. In the Add Site Binding dialog box, in the SSL Certificate field, select the name of the created certificate.
9. Click **OK**.

## Configure HTTPS Between the Java Web server and Platform Toolkit

To configure HTTPS between the Java Web server and Platform Toolkit:

1. Open the certmgr.mcs file located in the C:\Windows\SysWOW64 or C:\Windows\System32 directory. This will launch Certificate Manager.
2. Select **Trusted Root Certification Authority** ⇒ **Certificate**.
3. Locate the row containing the friendly certificate name in the Friendly Name column.
4. Right-click the row and select **All Tasks** ⇒ **Export**.
5. Accept all defaults in the Certificate Export Wizard except for the File to Export window. Enter a name for the certificate file in the File Name field. Click **Finish**.
6. From a command prompt, enter the following command:  
**"C:\Program Files (x86)\Java\jre6\bin\keytool" -import -keystore "C:\Program Files (x86)\Java\jre6\lib\security\cacerts" -aliasdracovht-file C:\Users\Developer\dracovht.cer**  
where:  
C:\Program Files (x86)\Java\jre6\bin = Location of Java used by Apache Tomcat  
C:\Program Files (x86)\Java\jre6 = Location of Java used by Apache Tomcat  
dracovht = Friendly name of the certificate  
C:\Users\Developer\dracovht.cer = Name of certificate file (refer to Step 5)
7. Enter the password (default password is changeit) for the keystore that will contain the imported certification file when prompted.
8. Reenter the password for the keystore that will contain the imported certification file when prompted.
9. Click **Yes** to trust the certificate when prompted.
10. Open the toolkit.properties file in a text editor. The default file location is C:\Virtual Hold.
11. In the #URL for PTK webservices section, change the http reference to https.
12. Ensure the name of the server on which the certificate was created is used in the #URL for the PTK webservices section. For example, the name could be *draco.qalab.local* in `https://draco.qalab.local/VHTPlatformWS-v4` or *intrepid* in `https://intrepid/VHTPlatformWS-v4`.
13. Restart Tomcat.

## Changing VXML Interaction Server Log Level

### Overview

Log messages will be written to the Apache Tomcat log file located at Tomcat 6.0\logs\Tomcat6\_stdout.YYYY-MM-DD.log (for Tomcat 7, substitute 7 for 6).

The VXML Interaction Server provides the following levels of logging.

Log Level	Information Written to the Log
ERROR	Errors
WARN	Errors and warnings
INFO (default)	Errors , warnings and informational output
DEBUG	Low level trace information

### Temporary Change

To change the log level for a short time (for example, when troubleshooting), perform the following steps in a web browser:

1. On the server where the VXML Interaction Server is installed, navigate to <http://server:port/warfile/-/logging>. The Current Log Level page is displayed.
2. Select the desired log level in the New Level field and click **Set**.
3. Change the level back when you are finished.

**Note:** The log level returns to the default setting when you restart the server or update the VXML Interaction Server application.

### Permanent Change

To change the level of detail in the logs permanently, perform the following steps in the Java Options:

1. On the Server where the VXML Interaction Server is installed, open the Tomcat6w.exe directory, usually found at C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin. The Properties window is displayed. (For Tomcat 7, substitute 7 for 6.)
2. In the Java Options section, add the **-Dorg.eclipse.vtp.loglevel=log\_level**. Valid values for *log\_level* are ERROR, WARN, INFO, and DEBUG.
3. Click **OK** or **Apply**.



4. Restart Apache Tomcat.

## Configuring HTTPS Support for Apache Web Server

Use these steps to configure HTTPS / SSL support for Apache Web Server for use with the VXML Interaction Server (VIS).

1. Locate the `httpd.conf` file and open it in a text editor.
2. Uncomment the following line:

```
| LoadModule ssl_module modules/mod_ssl.so
```

3. Add the "index.php" entry to the following section:

```
| <IfModule dir_module>  
| DirectoryIndex index.html index.php  
| </IfModule>
```

4. Uncomment the following line:

```
| Include conf/extra/httpd-ssl.conf
```

This will cause the `httpd-ssl.conf` file to be created in the `.../conf/extra` directory.

5. Save your changes.
6. Locate the `httpd-ssl.conf` file in the `.../conf/extra` directory and open it in a text editor.
7. Change the port number in the following line to the HTTPS port number (where **xxxx** is the port number).

```
| Listen xxxx
```

8. Change the port number in the following line to the HTTPS port number.

```
| <VirtualHost _default_:xxxx>
```

9. In the General Setup section, change the `ServerName` port number to the HTTPS port number.

```
| # General setup for the virtual host  
| DocumentRoot "c:/Apache24/htdocs"  
| ServerName server.test.local:xxxx  
| ServerAdmin admin@example.com
```

10. Save your changes.

## Deleting Expired Name Files

### Overview

The VXML Interaction Server application creates files while processing callbacks. Some of these files may not be deleted, so it is important to create a task to prevent expired files from building up in the system. To delete expired files, follow these steps for either Windows or Linux.

## Create and Schedule File Deletion Task (Windows 2008 Systems)

### Create Windows File Deletion Batch File

To remove expired temp folder, localhost\_access\_log and .wav files from Apache Tomcat, you will first create a batch file to delete these files, and then schedule the batch file using the Windows Task Scheduler.

To create a file deletion batch file:

1. Create a batch file.
2. Enter the following information into the batch file.

```
:: Location of the Tomcat Installation
set TomcatInstallationDirectory=C:\Program Files\Apache Software Foundation\Tomcat 7.0
:: Location of the NameFile Directory. This location should correspond to
:: com.virtualhold.toolkit.audiopath in the toolkit.properties file
set NameFileDirectory=%TomcatInstallationDirectory%\webapps\ROOT
27
Virtual Hold VXML Interaction Server Installation and Configuration Guide Configuration
:: Deletes localhost access logs that are older than 7 days forfiles /P
"%TomcatInstallationDirectory%\logs" /S /M localhost_access_log.* /D "-7" /C "cmd
/c del @PATH"
:: Deletes files in the temp directory (where partial name files are created)
forfiles /P "%TomcatInstallationDirectory%\temp" /S /M *.* /D "-8" /C "cmd /c del @PATH"
:: Deletes name files older than 8 days
forfiles /P "%NameFileDirectory%" /S /M *.wav /D "-8" /C "cmd /c del @FILE"
```

3. Configure the variables (in bold) appropriately and save the file.

### Schedule Windows File Deletion Batch File

To schedule execution of a file deletion batch file:

1. Start the Windows Task Scheduler (**Control Panel** ⇒ **Administrative Tools** ⇒ **Task Scheduler**).
2. Select **Create Basic Task...**

3. Enter a task name and description in the Create a Basic Task window.
4. Select **Daily** (the recommended value) in the Task Trigger window.
5. Select a start date and 1 day recur value (recommended) in the Daily window.
6. Enable **Start a program** in the Action window.
7. Select the batch file in the Program/Script field of the Start a Program window.
8. Read the summary and click **Finish** in the Summary window.

## Create and Schedule File Deletion Cron Job (Linux Systems)

Expired temp folder and localhost\_access\_log files from Apache Tomcat are by default managed by Linux and require no configuration changes. To remove expired .wav files, a cron job must be created and scheduled.

**Note:** The procedure for creating a cron job is different in each Linux distribution. The following procedure outlines the process performed at the command line of the Terminal application in an Ubuntu 12.04 Linux distribution.

To create a cron job that deletes .wav audio files older than eight days:

1. Log into the Linux system.
2. Start editing of the crontab by entering `sudo crontab -e`.
3. Enter the command to create the cron job in the following format:  
**`mm hh dd MM wd find path/file_type -mtime +x -exec rm -f {} \;`**

Where:

- **mm** = Minute the job will run. Valid entries \* and 0 through 59.
- **hh** = Hour the job will run. Valid entries \* and 0 (Midnight) through 23.
- **dd** = Day the job will run. Valid entries \* and 1 through 31.
- **MM** = Month the job will run. Valid entries \* and 1 through 12.
- **wd** = Weekday the job will run. Valid entries \* and 0 (Sunday) through 6.
- **path** = Path to files being deleted.
- **file\_type** = Files being deleted. \*.wav or config.\* for example.
- **x** = File age limit in days.

For example, the following job runs at 4:01 A.M. every day of every month and deletes all .wav files older than eight days from the /var/lib/tomcat7/webapps/ROOT folder.

```
01 04 * * * find /var/lib/tomcat7/webapps/ROOT/*.wav -mtime +8 -exec rm -f {} \;
```

4. Save the file.

## Installing the log4j Tomcat Logging Package

### Overview

Virtual Hold recommends installing the log4j logging package to provide a "rolling" log for Apache Tomcat.

### Apache Tomcat 7

To install log4j on Apache Tomcat 7:

1. Navigate to the root Tomcat directory (for example, C:\Program Files\Apache Software Foundation\Tomcat 7.0).
2. Create a backup of this directory.
3. Download the Tomcat-6-7-8-LogRollingFiles4-24-2014 collection from VHT Insight at [https://insight.virtualhold.com/kb/Tomcat\\_7\\_upgrade\\_and\\_Log4j\\_deployment\\_instructions](https://insight.virtualhold.com/kb/Tomcat_7_upgrade_and_Log4j_deployment_instructions).
4. Copy/Replace/edit the following files from the log4j.zip collection accordingly:

\*\*\*\*Bin directory\*\*\*\*

\bin\tomcat-juli.jar

\*\*\*\*Lib directory\*\*\*\*

\lib\log4j.properties

\lib\log4j-1.2.16.jar

\lib\tomcat-juli.jar

\lib\tomcat-juli-adapters.jar

**Note:** The default setting is to generate a new log file when the current log file reaches 10MB in size. Also by default, the 200 most recent log files will be kept (i.e., when the 201st file is generated, the first file will be deleted, and so on). To change these settings, edit the following lines in the log4j.properties file:

```
log4j.appender.CATALINA.MaxFileSize=10MB
```

```
log4j.appender.CATALINA.MaxBackupIndex=200
```

5. Navigate to the conf folder in the root of the Tomcat directory.
6. Open the context.xml file in a text editor.
7. Add `SwallowOutput="true"` to the context tag. **Be sure to retain the header `useHttpOnly="false"` if it exists already** - for example:  
`<Context swallowOutput="true" useHttpOnly="false">`
8. Save the context.xml file.
9. Restart Tomcat.

## Load Balancing and Failover for the VXML Interaction Server

### Overview

Load balancing directs one call to one server and the next call to another server in a round-robin fashion. This functionality divides the load between the two servers. Failover provides an additional server for use if the first server fails.

To accomplish load balancing or failover when you use the VXML Interaction Server, you must store the name files in a shared directory and enable the applications for this functionality.

#### Storing the Name Files in a Shared Directory

If you are using multiple VXML Interaction Servers, you can share the name files across all of the servers instead of loading them on each server. This configuration might be useful anytime you are using multiple VXML Interaction Servers; for example, for load balancing, or in an enterprise environment.

To use a shared directory for name files, you must create the shared directory, create the recordings.vxml file, and revise the toolkit.properties file. Perform the following steps:

1. Create the name files directory in a location that can be accessed by all of the VXML Interaction Servers.
2. Create the recordings.vxml file. Add the path to the directory where the name files are stored. The file should contain the following text:  
`<Context docBase="absolute path to name recordings directory"> </Context>`
3. Place the file in the following location on each of the servers that contain a VXML Interaction Server:  
*tomcat dir/conf/Catalina/localhost* on one of the servers.
4. Update the following lines in the toolkit.properties file to point to the location of the directory where the name files are stored:
  - `com.virtualhold.toolkit.audiopath=C:/Program Files (x86)/Apache Software Foundation/Tomcat 7.0/webapps/ROOT/NameFiles/`
  - `com.virtualhold.toolkit.webaudiopath=http://servername:port/recordings`
5. Copy the toolkit.properties file to the rest of the servers that contain VXML Interaction Servers.

#### Using Two VXML Interaction Servers with AVP

If you want to use two VXML Interaction Servers for load balancing or failover, you must enable the feature for the applications in Avaya Voice Portal (VPMS).

Perform the following steps:

1. Navigate to the inbound application under **Applications**.
2. In the URL, select the appropriate option: **Load Balancing** or **Failover**.
3. Enter the URL for the second server.



4. Repeat these steps for the outbound application.

## Media Server Latency

### Overview

Some latency may occur when playing voice files from an external media server. This will affect the VXML Interaction Server in different ways, depending on whether the response is received within the voice browser's timeout setting.

- If the latency is less than the browser timeout setting, the customer will experience a period of silence intermittently during the call flow when the latency occurs. Prompts will not be skipped and will still play in order.
- If the latency is greater than the browser timeout setting, the customer will hear silence for the duration of the latency issue. The prompts which would normally have been played during this period will be skipped. VIS will continue to play the rest of the prompts after the latency issue is resolved.

There is no media server failover during either scenario. A failover is only triggered if the initial connection cannot be made to the media server.

## Media Server Logging in Apache Tomcat

### Overview

Custom media log messages are written to the Apache Tomcat log file. The default log location is Tomcat 6.0\logs\Catalina.x.YYYY-MM-DD.log, where x may be a number from 1-100 (or however many log files you choose to set up) (for Tomcat 7, substitute 7 for 6).

Upon connection, the following messages are written to the log:

```
Loading Configuration from C:\VirtualHold\toolkit.properties
Adding external media server: [path to server]
```

When voice prompts are loaded, messages similar to the following are written to the log:

```
Setting brand to: Default
Set brand to /Default
Setting language to: English
Set language to English
Comparing to: English
Setting library to: CallScript_2
Setting media library to: CallScript_2
resolving resource: /project/Media Libraries/CallScript_2/.library
```

When a failure occurs, messages similar to the following are written to the log:

**Note:** Failure messages will be logged depending on the setting in toolkit.properties. Refer to [toolkit.properties for External Media Files](#).

```
Unable to connect to external media server @ http://10.10.1.61:8180/voices/VHT_FrenchCanadian/
Unable to connect to external media server @ http://10.10.1.61:8180/voices/VHT_English/
java.net.ConnectException: Connection refused: connect
Unable to connect to external media server @ http://10.10.1.61:8180/voices/VHT_SpanishNA/
    at java.net.PlainSocketImpl.socketConnect(Native Method)
    at java.net.PlainSocketImpl.doConnect(Unknown Source)
    at java.net.PlainSocketImpl.connectToAddress(Unknown Source)
    at java.net.PlainSocketImpl.connect(Unknown Source)
    at java.net.SocksSocketImpl.connect(Unknown Source)
    at java.net.Socket.connect(Unknown Source)
    at java.net.Socket.connect(Unknown Source)
    at sun.net.NetworkClient.doConnect(Unknown Source)
    at sun.net.www.http.HttpClient.openServer(Unknown Source)
    at sun.net.www.http.HttpClient.openServer(Unknown Source)
    at sun.net.www.http.HttpClient.<init>(Unknown Source)
    at sun.net.www.http.HttpClient.New(Unknown Source)
```

```
at sun.net.www.http.HttpClient.New(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.getNewHttpClient(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.plainConnect(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.connect(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown Source)
at org.eclipse.vtp.framework.engine.ResourceGroup$1.run(ResourceGroup.java:97)
at java.lang.Thread.run(Unknown Source)
java.net.ConnectException: Connection refused: connect
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.PlainSocketImpl.doConnect(Unknown Source)
at java.net.PlainSocketImpl.connectToAddress(Unknown Source)
at java.net.PlainSocketImpl.connect(Unknown Source)
at java.net.SocksSocketImpl.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at sun.net.NetworkClient.doConnect(Unknown Source)
at sun.net.www.http.HttpClient.openServer(Unknown Source)
at sun.net.www.http.HttpClient.openServer(Unknown Source)
at sun.net.www.http.HttpClient.<init>(Unknown Source)
at sun.net.www.http.HttpClient.New(Unknown Source)
at sun.net.www.http.HttpClient.New(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.getNewHttpClient(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.plainConnect(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.connect(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown Source)
at org.eclipse.vtp.framework.engine.ResourceGroup$1.run(ResourceGroup.java:97)
at java.lang.Thread.run(Unknown Source)
java.net.ConnectException: Connection refused: connect
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.PlainSocketImpl.doConnect(Unknown Source)
at java.net.PlainSocketImpl.connectToAddress(Unknown Source)
at java.net.PlainSocketImpl.connect(Unknown Source)
at java.net.SocksSocketImpl.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at sun.net.NetworkClient.doConnect(Unknown Source)
at sun.net.www.http.HttpClient.openServer(Unknown Source)
at sun.net.www.http.HttpClient.openServer(Unknown Source)
at sun.net.www.http.HttpClient.<init>(Unknown Source)
at sun.net.www.http.HttpClient.New(Unknown Source)
at sun.net.www.http.HttpClient.New(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.getNewHttpClient(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.plainConnect(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.connect(Unknown Source)
at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown Source)
at org.eclipse.vtp.framework.engine.ResourceGroup$1.run(ResourceGroup.java:97)
at java.lang.Thread.run(Unknown Source)
```

The voice prompts themselves are logged in the browser log for GVP, CVP, or AVP (or other voice platform used), not the Tomcat log.

## Migrating from VXML Interaction Server 4.3.2 to 5.0.x

### Overview

Moving from VXML Interaction Server 4.3.2 to 5.0 or higher involves loading a new .war file, editing the toolkit.properties file, and deploying voice prompts to separate media servers (if desired).

### Before Migration

Before migrating to Release 5.0 or later of the VXML Interaction Server:

1. Stop Apache Tomcat
2. Create a backup of the current toolkit.properties and .war files, and then rename both for backup purposes.

### Steps to Migrate

To migrate to Release 5.0 or later of the VXML Interaction Server:

1. Ensure the VXML Interaction Server is completely installed.
2. Restart Apache Tomcat.

### If Rollback is Needed

To rollback to the previous release of VXML Interaction Server:

1. Stop Apache tomcat
2. Change the name of the backup files you made in the Before Migration section above to the names the system is provisioned for.
3. Delete the version 5.0.x .war file. and replace it with the backup .war file.
4. Replace the toolkit.properties file with the backup file.
5. Delete all the log files from within Tomcat 6.0\logs (for Tomcat 7, substitute 7 for 6).
6. Delete the directory matching the name of the deleted .war file from Tomcat 6.0\work\catalina\localhost (for Tomcat 7, substitute 7 for 6).

## Troubleshooting External Media Server Issues

### Overview

Use these steps to troubleshoot VXML Interaction Server media server, voice prompt, and latency issues.

### Media Server Issues

Confirm that Apache Tomcat is still running by looking for the icon in the system tray, or by checking Windows services (**Control Panel > Administrative Tools > Services**).

Confirm that each media server is running properly by checking the Tomcat log. The default log location is Tomcat 6.0\logs\Tomcat6\_stdout.YYYY-MM-DD.log (for Tomcat 7, substitute 7 for 6). Refer to [Media Server Logging in Apache Tomcat](#) and [Changing VXML Interaction Server Log Level](#).

Look for any failure messages such as "Unable to connect to external media server..." If you see a failure message, check the path to the server by looking in the toolkit.properties file. Look for external.mediaserver.1=[path to server].

- Apache Tomcat and Apache Web Server: Test the deployed files by browsing to this path in a web browser. The browser should be able to locate the folder. If the browser displays an error, make sure the files have been deployed to the correct location and Tomcat/Apache Web Server is running properly.
- Microsoft IIS: Test the deployed files by right-clicking the virtual directory and selecting **Manage Virtual Directory > Browse**. The browser should be able to locate the folder. If the browser displays an error, make sure the virtual directory is configured properly.
- Refer to [Deploying External Media Files](#) for more details.

Also check toolkit.properties to see whether the servers are used in balanced mode or failover mode. You may need to change to the other mode. Refer to [Configuring toolkit.properties for External Media Files](#).

If a media server fails (i.e., the VXML Interaction Server (VIS) cannot connect to it), there is no "timeout" on the failed server. VIS will keep attempting to connect to the failed server. The behavior varies, depending on whether you are using balanced mode or failover mode:

- Failover: If the first media server is down, the browser will keep trying the failed server for each subsequent call. If the second or other media server is down, all calls will be hitting the first media server.
- Balanced: If one of the media servers is down, the browser will still attempt some percentage of the calls on the failed server, but not all the calls. We recommend using balanced mode over failover mode for this reason.

### Voice Prompt Issues

Check the browser log for GVP, CVP, or AVP (or other voice platform). You will see messages for the specific prompts being played.



If an incorrect prompt is played, check the structure and contents of the voices folder. Make sure all the required files are present in the correct folders and the names are spelled correctly. Refer to [Customizing External Media Files](#).

Also, be sure the Default folder and each custom folder contains an empty .library file (size 0 KB). If needed, the .library file from the Default folder can be copied and pasted.

## Latency Issues

Network latency may cause periods of silence for callers as VIS waits for a response from the media server. This will cause the voice prompts to be delayed or possibly skipped. Refer to [Media Server Latency](#).