



Design Implications for VoiceXML Server

This chapter cover the following topics:

- [What is VoiceXML over HTTP?](#), page 13-1
- [Multilanguage Support](#), page 13-2
- [Differences in the Supported Web Application Servers](#), page 13-2
- [Where to Install CVP Studio](#), page 13-3

What is VoiceXML over HTTP?

Communication between VoiceXML server and the VoiceBrowser is based on request-response cycles using VoiceXML over HTTP. VoiceXML documents are linked together by using the Uniform Resource Identifiers (URI), a standardized technology to reference resources within a network. User input is carried out by web forms similar to HTML. Therefore, forms contain input fields which are edited by the user and sent back to a server.

Resources for the Voice Browser are located on the VoiceXML server. These resources are VoiceXML files, digital audio, instructions for speech recognition (Grammars) and scripts. Every Communication process between the VoiceXML browser and Voice Application has to be initiated by the VoiceXML browser as a request to the VoiceXML server. For this purpose, VoiceXML files contain Grammars which specify expected words and phrases. A Link contains the URL which refers to the Voice application. The browser connects to that URL as soon as it recovers a match between spoken input and one of the grammars.

So when gauging VoiceXML server performance, key aspects to consider are:

- Network bandwidth between Web application server and the VoiceGateway and QOS.
Refer to [Bandwidth Provisioning and QoS Considerations](#), page 8-1 for more details.
- Performance on the VoiceXML Server
CVP Bill of Materials (BOM) requires the MCS-7845 as a VoiceXML server. Adequate performance is required on the server side to respond to VoiceXML over HTTP requests.
- Use of pre-recorded Audio vs. Text to Speech
Good Voice User Interface applications tend to use pre-recorded audio files wherever possible. Recorded audio sounds much better than TTS. Pre-recorded Audio file quality needs to be designed such that it does not impact download time and browser interpretation. Make recordings in 8-bit Mu law 8Khz format.
- Audio File Caching

Make sure the Voice gateway is set to cache Audio content prevents delays in having to download files from the media source.

Refer to the Section titled Gateway Prompt Caching Considerations for more details on Prompt Management on Supported Gateways

- Use of Grammars

A voice application, like any user-centric application, is prone to certain problems that might only be discovered through formal usability testing, or observation of the application in use. Poor speech recognition accuracy is one type of problem common to voice applications, and a problem most often caused by poor grammar implementation. When users mispronounce words or say things that the grammar designer does not expect, the recognizer cannot match their input against the grammar. Poorly designed grammars containing many difficult-to-distinguish entries also results in many misrecognized inputs leading to decreased performance on the VoiceXML server.

Grammar tuning is the process of improving recognition accuracy by modifying a grammar based on an analysis of its performance.

Multilanguage Support

The IOS Voice Browser or the MRCP specification does not impose restrictions on support for Multiple Languages. However, there might be restrictions on the ASR/TTS server; check with your preferred ASR/TTS vendor on their support for your languages before preparing a multi-lingual application.

Programatically, there is a method where you can dynamically change the ASR server value using a cisco property com.cisco.asr-server in the VoiceXML script. This property overrides any previous value set by the VoiceXML script.

Differences in the Supported Web Application Servers

From a very high level perspective, IBM WebSphere Application Server (www.ibm.com/websphere) is a complete J2EE application server environment complete with an Administration console and connection pooling. However, Tomcat (<http://tomcat.apache.org/>) is a simple and a basic environment with a Servlet Engine and a Java Server Pages engine only. The decision to use Tomcat or Websphere Application Server depends on the customer's current enterprise infrastructure requirements. In many cases, Tomcat is more than sufficient, but if a customer already has WebSphere infrastructure and management capabilities, or has a preference for WebSphere in general, he should use it for CVP.

Performance tests conducted on the web application server showed only slight variations in the Processor performance between the two Web Application Servers using metrics such as:

- Impact of call volume
- Impact of application size
- Impact of Application complexity

Both Tomcat and Websphere Application server running CVP VoiceXML can support up 500 simultaneous calls per 7845 physical box.

Where to Install CVP Studio

CVP Studio is an Integrated Development Environment (IDE). As in the case of any IDE, the studio needs to be installed in a setup that is conducive for development, such as workstations that are used for other software development or business analysis purposes. Since the CVP studio is Eclipse-based, many other development activities such as writing Java programs or building object models can be migrated to this tool so that developers/Analysts have one common utility for most of their development needs.

For non-production systems, the CVP studio can be installed in conjunction with the CVP VoiceXML server. If the intent is to only test applications in a non load scenario, the co-resident configuration is acceptable.

