WebSphere Voice Server for Multiplatforms 5.1

Contents

Chapter 1. Product overview

These topics describe the product and its surrounding business and technology context.

Speech in e-business

Speech technology has come of age, and IBM WebSphere Voice Server 5.1 draws on more than 30 years of worldwide speech research and technology expertise to bring you state-of-the-art functionality.

What does speech offer your e-business? Simply put, speech technology offers customer satisfaction.

You have heard the common approach to telephone-based customer service: "For x, press 1. For y, press 2. For z, press 3." This clunky form of interaction relies on keypad presses and limited numerical user input, creating a confusing maze of menus and a less-than-satisfactory customer experience. At the same time, using live operators for customer service results in inconsistent service quality and significant expense.

Speech technology is the future of e-business because it enables more natural, intuitive, and engaging customer service for less cost. Implementing speech technology means your business is always accessible to your customers.

Speech applications often provide return on investment in as few as three to six months. Speech technology means that your customer service representatives can handle the complex service tasks they are best equipped, not simple, repetitive customer transactions. With speech, simple customer service needs might be handled by technology to promote a more cost-effective customer service strategy in your e-business. In addition, a well-designed speech user interface can improve customer satisfaction and loyalty over Web site self-service by giving your customers the flexibility to receive their service anywhere, anytime from their telephone, the most common personal technology.

The numerous benefits of speech technology for e-business include:

- Interaction with callers is easier and more natural.
- Menus can be eliminated or flattened, for more subtle and intuitive navigation.
- Call durations can be minimized, meaning less cost per transaction.
- Interaction with customers can occur 24 hours a day, 7 days a week.
- Customers interact with your business using their telephone or cell phone, resulting in continuous access to the customer base regardless of their location.
- Individuals with physical or perceptual disabilities might have greater access to e-business services.
- Enterprise branding is extended to a new channel – phone-based interaction.
- Overall customer service costs are decreased.
- Return on investment for speech application development often occurs in less than a year.
- Opportunities for integrating and streamlining business processes arise as speech applications are developed.
- New business opportunities can arise using speech technology.

These benefits combine to increase customer satisfaction and loyalty, meaning that speech technology is central to effective, efficient e-business on demand.

Brief introduction to speech

This section is a quick primer on speech technology, to get you ready to work with speech technologies and welcome this new form of interaction in your business environment.

Speech refers to the recognition and production of spoken language. Speech technology, at its most basic, includes one technology for recognition, known as automatic speech recognition (ASR), and a second technology for the production of spoken language. Technology-generated speech, or the “system voice,” is achieved in one of three ways:

- Using text-to-speech (TTS) technology, which creates spoken language from written language
- Using recorded human voice segments (prompts)
- By combining TTS and recorded voice segments

Currently, combinations of TTS and recorded voice segments provide the most flexible and cost-effective solutions for the majority of speech applications.

The accuracy of ASR (did it accurately recognize what the user said?) and the pleasantness and naturalness of the system voice largely determine the quality of speech applications. Other elements of quality are determined by the speech interface design of the application, such as the flow of the dialogue and use of language in prompts.

How does ASR work?

ASR enables a server to recognize general, naturally flowing utterances from a wide variety of users. It recognizes the caller’s answers to move along the flow of the call.

ASR technology converts acoustic waves from a user‘s spoken words to a form that is usable by a speech application. This means that speech input can be accepted from a user and processed in the same way as dual tone multi frequency (DTMF) input, text, or numeric data. The outcome of a correctly recognized user utterance is determined by the call flow of a speech application.

Several important terms are associated with ASR:

Voice model
A set of parameters for the language to be recognized, including the acoustic model, the numeric model of how the language sounds when spoken.

Baseform
A description of what the words should sound like (the pronunciation). The baseforms can be modified by an application developer to better match users‘ “actual” responses in a speech application.

Grammar
The specification of the words and phrases that are active and can be recognized by the ASR engine at a given time. Grammars can be created dynamically or at application development time, and can be loaded, unloaded, activated and deactivated under the voice application control.

Lexicon
The set of words in a given language or system. A lexicon refers to the set of files that comprise the vocabulary of an application or system. Lexicons are used by the ASR engine to increase its recognition accuracy.

The following is a sample of an ASR lexicon:

```
<?xml version='1.0' encoding='utf-8'?>
<lexicon version='1.0' xml:lang='en-US' alphabet='x-ibmasr' case-sensitive='false'>
 <import uri='sourcelexicon.xml'/>
 <lexeme>
    <spelling>preferred</spelling>
```

```
      <phoneme>P R AX F ER DD</phoneme>
      <phoneme>P R IX F ER DD</phoneme>
    </lexeme>
   <lexeme>
       <spelling>colour</spelling>
       <spelling>color</spelling>
       <phoneme>K AH L AXR</phoneme>
   </lexeme>
   <lexeme>
      <spelling>IEEE</spelling>
      <sounds-like>I triple E</sounds-like>
    </lexeme>
  </lexicon>
```

How does TTS work?

TTS is typically used as part of the system voice in speech applications. It often is used for variable items in a dialogue, such as a user address or name, specific dialogue items that are dynamic and thus cannot be predicted by the application developer.

In a working speech interaction, TTS segments can be "spliced" into prompts spoken by a professional voice talent to create unique messages for each user. For example, the sentence "Your balance is $7513.45" can use a recorded voice talent for the static "your balance is" and TTS for "$7513.45" because the amount of money in each user's account is dynamic information and too variable to record professionally in a cost-effective way.

Several important terms are associated with TTS:

Lexicon
The set of words in a given language or system. A lexicon refers to the set of files that comprise the vocabulary of an application or system. Lexicons are used by the TTS engine to enhance the quality of its pronunciation.

Baseform
A description of what the words should sound like (the pronunciation).

Speech Synthesis Markup Language (SSML)
A markup language that helps application developers using TTS create standardized synthesizable output from the speech engines.

What does WebSphere Voice Server do?

WebSphere Voice Server provides the ASR and TTS resources that enable speech-based interaction. A speech application (VoiceXML content) provides the logic and flow of dialogue. A VoiceXML browser provides access to the speech application (VoiceXML content) from a Web server. When these elements are combined, the speech interface is perceived as a series of system utterances that require user responses. Several system-user-system responses make up a conversational interaction.

The WebSphere Voice Server allows system administrators to configure, monitor, and troubleshoot voice server resource usage in deployed speech applications

Related information

WebSphere Voice family

Speech Synthesis Markup Language Version 1.0

Speech Recognition Grammar Specification Version 1.0

Product description

IBM WebSphere Voice Server for Multiplatforms Version 5.1 offers a Media Resource Control Protocol (MRCP) server for speech technology.

WebSphere Voice Server for Multiplatforms Version 5.1

IBM WebSphere Voice Server for Multiplatforms 5.1 confirms IBM's commitment to support, adopt, and drive open standards. With WebSphere Voice Server 5.1, IBM moves from providing proprietary application program interfaces (APIs) to its speech technology to the industry standard – the MRCP interface. WebSphere Voice Server also builds on the base of other IBM WebSphere products to provide scaling, load balancing, failover, recovery, systems management, logging, tracing, and problem determination consistent with the IBM WebSphere family. WebSphere Voice Server leverages the ease of use and administration capabilities of IBM's WebSphere Application Server, the leading application server in the industry, with proven capabilities for scaling and performance.

For additional information on WebSphere Application Server, see the WebSphere Application Server Network Deployment 5.1 Information Center

ASR support

Automatic speech recognition (ASR) is the technology that matches spoken input to the expected input (grammar) in a speech application. WebSphere Voice Server offers four languages with significantly improved performance in noisy environments, especially in reducing false barge-ins. These languages include:

- Japanese
- Simplified Chinese
- U.K. English
- U.S. English

TTS support

TTS converts written text into spoken words and often provides the voice of a speech application. WebSphere Voice Server provides improved synthesis and signal processing algorithms. TTS supports:

- Speech Synthesis Markup Language (SSML)
- SSML lexicons
- Persistent lexicons
- SSML embedded audio
- <voice-name> attribute

TTS also offers several advanced TTS male and female voices for the following languages:

- U.S. English (one male voice, two female voices)
- U.K. English (one male voice, one female voice)
- Japanese (one male voice, one female voice)
- Simplified Chinese (one female voice)

MRCP server

The Internet Engineering Task Force (IETF) defines MRCP as a protocol for controlling media service resources, such as a speech recognizer or a speech synthesizer. The following figure shows the relationship of WebSphere Voice Server to a partner platform and the application server.

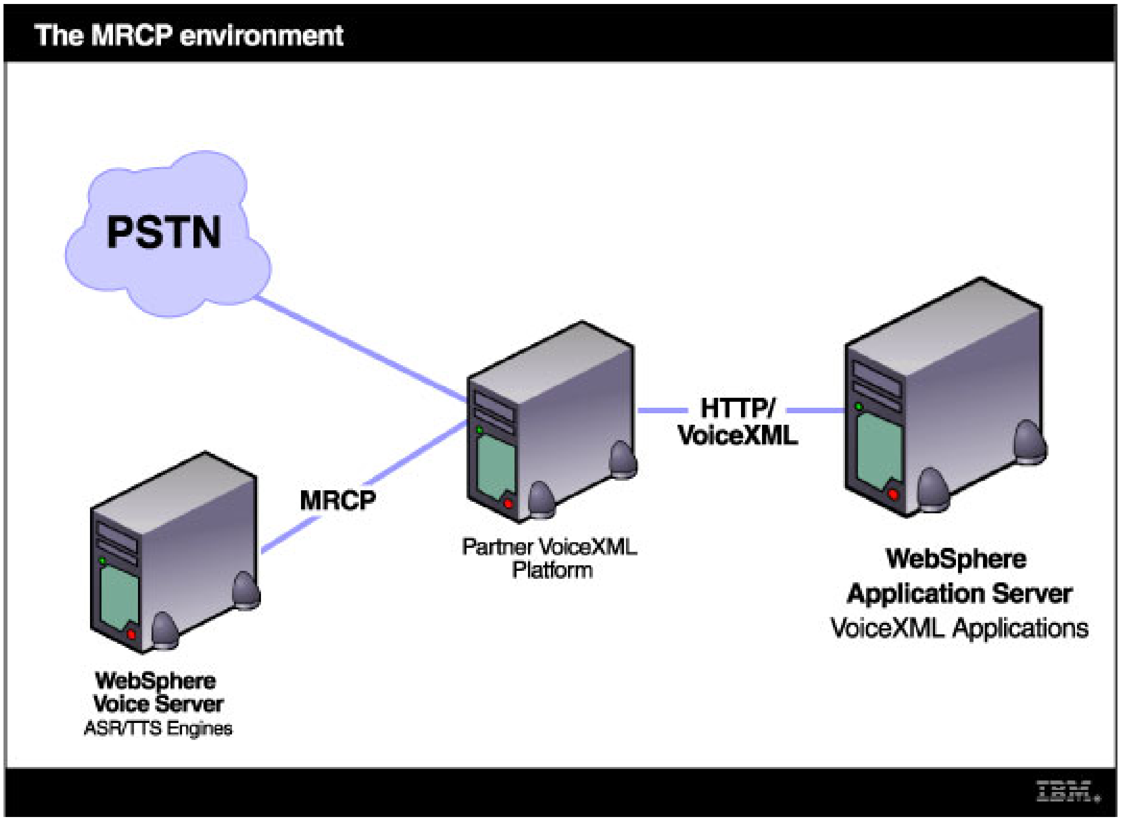


Figure 1. The MRCP environment

The WebSphere Voice Server base provides the features of the MRCP server that conform to draft four of the MRCP version 1 specification definition. When you add a language support package, the MRCP server acts as a recognizer, providing a standard interface into the ASR resources. The Concatenative TTS package supports the synthesizer functions defined in the MRCP specifications. When you deploy WebSphere Voice Server, you can install both recognition and synthesis resources on the same physical server, or on separate servers. Scaling of the MRCP server is available using the WebSphere Network Dispatcher. The Network Dispatcher enables multiple physical servers to present a single server name to which the MRCP clients can connect.

Features

By leveraging the features and functions that an administrator uses for an application server, WebSphere Voice Server provides administrative ease of use to voice administrators.

For e-businesses, speech allows an alternate means of access to business services and transactions, especially for individuals with disabilities. Using an architectural foundation provided by WebSphere Application Server, IBM's premier product for e-business technology, the Voice Server also provides extensive administrative and performance benefits for complex speech technology architectures, including the following.

Accessibility

Accessibility features enable users with physical disabilities, such as restricted mobility or limited vision, to operate software products.

IBM WebSphere Voice Server for Multiplatforms is a voice application enabler, which appeals to the major user groups impacted by speech technology – system administrators, application developers, and application end users. WebSphere Voice Server version 5.1 provides the following accessibility features for each user group:

System Administrators

- Provides accessible administration and monitoring interfaces for speech technology resources.
- Offers text through standard system function calls or through an open application programming interface (API), which supports interaction with assistive technology.
- Provides accessible documentation.

Telephone Callers (End Users)

- Offers the capability of providing people with visual, manual, and/or mobility impairments access to business data and services using auditory user interfaces.
- Helps businesses comply with Section 508 of the Americans with Disabilities Act.

Application Developers

- Provides an accessible development environment for speech applications.

The administrative console is the primary interface for interacting with the product. This console is displayed within a standard Web browser. By using an accessible Web browser, such as Microsoft Internet Explorer, administrators are able to:

- Use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen
- Use voice recognition software, such as IBM ViaVoice, to enter data and to navigate the user interface
- Operate features by using the keyboard instead of the mouse

One can configure and administer product features by using standard text editors and scripted or command line interfaces instead of the graphical interfaces provided.

When appropriate, the documentation for specific product features contains additional information about the accessibility of the features.

Administrative ease of use

One feature of WebSphere Voice Server version 5.1 is its administrative console, a GUI control panel for administration and monitoring of speech resources.

The console provides views of overall system status and detailed status of automatic speech recognition (ASR) and text-to-speech (TTS) resources, as well as general resources (system status, free memory). System administrators can obtain “at-a-glance” knowledge of their system by reviewing a single interface.

In addition to a speech-centric view of voice servers, administrators familiar with WebSphere Application Server can also view their system from the Application Server perspective, for comprehensive system understanding.

The console is also a centralized configuration tool. Administrators can use the console panel to configure the ASR and TTS engines, preloaded grammars, lexicons, and installed voices.

Finally, for administrators who prefer to administer using scripts, WebSphere Voice Server provides several example Jacl scripts to help you quickly learn the skills to produce your own unique administration code.

Continuous access

Your speech applications provide customer service and business transactions, save money, and streamline e-business processes. WebSphere Voice Server‘s ability to support complex architectures ensures that your business will always be available to your customers, due to the following features.

Availability and failover support
Avoid a single point of failure and maximize system availability by ensuring that the topology has some degree of process redundancy. High-availability topologies typically involve horizontal scaling across multiple machines.

Load balancing capability
A Network Dispatcher server performs intelligent load balancing to determine where to send a TCP/IP request. It can direct client RTSP/MRCP requests to available voice servers, bypassing any that are offline. Another server can back up the Network Dispatcher server, to eliminate it as a single point of failure. Workload management of voice servers also improves availability and failover support. Failover support distributes client RTSP/MRCP requests to the remaining servers, which ensures continued client access without significant interruptions.

Maintainability
The ability to update hardware and software is another important consideration for many speech technology infrastructures. WebSphere Voice Server allows you to create multiple Network Dispatcher clusters (horizontal scaling) that can make a system easier to maintain because you can take individual machines offline without interrupting other machines that are running a speech application. Although taking a system offline in a cluster can reduce overall throughput, for those applications that must continue running in the event of maintenance, WebSphere Voice Server‘s flexibility allows you to design your system appropriately.

Dynamic configuration
Modifying the system configuration without interrupting its operation enhances the manageability and flexibility of the system. For instance, administrators can add or remove cluster members to handle variations in the client load, change server characteristics and propagate the changes to its cluster members, temporarily stop servers for maintenance, and so forth.

Scalability

Adding more machines increases processing power, which is important for rapid exchanges in speech applications.

WebSphere Voice Server provides horizontal scalability, adapting readily to a greater or lesser intensity of use, volume, or demand. Horizontal scaling creates additional WebSphere Voice Server processes on multiple physical machines to take advantage of additional processing power. You can use WebSphere Application Server Edge (Network Dispatcher) with horizontal scaling to provide a single system view of multiple physical machines.

Rapid problem determination

WebSphere Voice Server includes comprehensive messages and alarms to keep you fully informed about system status so that you can quickly target and resolve problems using the administrative console. All trace and log messages are now in a centralized repository.

Troubleshooting tools, including the Log Analyzer and Tivoli Performance Viewer, enable real-time analysis and system monitoring. If you require support, a Collector Tool gathers system information for support personnel. These features expedite problem determination and resolution, saving you time and money.

Security

IBM WebSphere Voice Server uses the security infrastructure and mechanisms provided by WebSphere Application Server.

WebSphere Application Server allows administrators to protect a server using two security features: Global Security and Java 2 Security. These security features are managed through the Security panel on the administrative console.

Important: Java 2 Security cannot be enabled in order for WebSphere Voice Server to function properly. If you need security, enable Global Security in the WebSphere Voice Server administrative console. Do *not* select the option to enforce Java 2 Security.

For detailed information about the WebSphere approach to security, refer to Welcome to Security in the WebSphere Application Server information center.

Product GUIs and tools

IBM WebSphere Voice Server provides a variety of graphical and non-graphical user interfaces, to provide an end-to-end speech solution for your e-business.

Voice Toolkit for WebSphere Studio

An important part of any successful speech development effort, IBM Voice Toolkit for WebSphere Studio offers a full-function development environment for VoiceXML applications. The Voice Toolkit includes the drag-and-drop Call Flow Builder tool, which allows rapid development of your speech applications for usability evaluation and iterative design. For more information, see Voice Toolkit for WebSphere Studio.

Tools for installing

WebSphere Voice Server includes easy-to-use tools to get your voice server installed and running with maximum ease of use.

First Steps

A comprehensive beginning point from which you can access the information center, test your installation, and start or stop the WebSphere Voice Server. See firststeps command.

Launch Pad

A graphical interface for starting the product installation. It also provides links to helpful install information. See Using the Launchpad to start the installation.

Installation wizard

A graphical interface that leads you through the WebSphere Voice Server installation in a series of steps.

Tools for developing applications

The voiceTest tool provides you assistance as you develop speech applications and prepare them to work with WebSphere Voice Server. The voiceTest tool is a command-line interface that allows basic interaction with the WebSphere Voice Server MRCP server. The tool is used for installation verification, grammar testing, and ASR and TTS engine testing.

Tools for troubleshooting

WebSphere Voice Server also provides tools to help you with rapid problem determination for your voice system.

Administrative console

The administrative console is a graphical interface for performing deployment and system administration tasks. For more information, see Using the administrative console.

Collector tool

The Collector Tool gathers information about your WebSphere Voice Server installation (including WebSphere Application Server) and packages it in a JAR file that can be sent to IBM Customer Support to assist in problem determination and analysis. For more information, see Gathering information with the Collector tool.

Log Analyzer

The Log Analyzer takes one or more service or activity logs, merges all of the data, and displays the entries. Based on its symptom database, it analyzes and interprets the error conditions in the log entries to help you debug problems. Log Analyzer has a special feature enabling it to download the latest symptom database from the IBM Web site. For more information, see Log Analyzer.

Tivoli Performance Viewer

Tivoli Performance Viewer is a GUI that allows you to monitor system performance in real time. This useful tool can be used with the “moment-in-time” monitoring provided by the WebSphere Voice Server administrative console. For more information, see Tivoli Performance Viewer.

Relationship to other versions of WebSphere Voice Server products

Even though the WebSphere Voice Server for Multiplatforms Version 5.1 package includes WebSphere Voice Server Version 4.2 for AIX, there is no relationship between the two Voice Server versions.

WebSphere Voice Server V5.1 is not compatible with WebSphere Voice Response or with WebSphere Voice Server 4.2. Only WebSphere Voice Server V4.2 is compatible with WebSphere Voice Response.

WebSphere Voice Server V5.1 does not include the WebSphere Voice Server Speech Technologies, WebSphere Voice Server/Windows or Cisco/Dialogic products. WebSphere Voice Server V5.1 supports only the Linux operating system.

Refer to the WebSphere Voice Server product page to find the current list of platforms supported.

WebSphere family overview

IBM provides a family of WebSphere products that create a reliable, standards based infrastructure for your e-business.

WebSphere Voice Server is closely related to WebSphere Application Server, which is the architectural foundation of IBM‘s speech technology. To learn more about WebSphere Application Server and increase your knowledge of WebSphere Voice Server‘s advanced capabilities, refer to the WebSphere Application Server Network Deployment 5.1 information center. IBM‘s other WebSphere products complement IBM speech technology for comprehensive e-business solutions.

WebSphere foundation and tools

In addition to WebSphere Application Server, two foundation products provide scalability and productivity for rapidly and reliably growing e-business applications:

- WebSphere MQ

  This software exchanges information among more than 35 platforms with assured delivery. For more information, see http://www.ibm.com/software/ts/mqseries/.
- WebSphere Studio Site Developer

This software offers an integrated development environment based on a common workbench technology. Based on open standards, these tools enable exploitation of the WebSphere runtime environment. For more information, see http://www.ibm.com/software/awdtools/studiositedev/.

WebSphere reach and user experience products

The following products collaborate to help you achieve customer loyalty by extending and personalizing user experiences:

- WebSphere Portal

  This software accesses widespread and diverse data sources from anywhere, anytime, by anyone you allow. For more information, see http://www.ibm.com/software/webservers/portal/.
- WebSphere Everyplace

  This is a software infrastructure that supports mobile solutions, addressing the challenge of extending e-business applications to mobile devices. For more information, see http://www.ibm.com/software/pervasive/.
- WebSphere Commerce

  These powerful solutions are designed to handle the broad range of challenges encountered when selling in Business-to-Business (B2B) and Business-to-Consumer (B2C) environments. For more information, see http://www.ibm.com/software/webservers/commerce/.

WebSphere business integration products

The following products collaborate to help you achieve business agility by integrating applications and automating business processes:

- WebSphere Business Integration

  This software creates the nimble infrastructure needed to support the business imperatives of your dynamic enterprise. For more information, see http://www.ibm.com/software/integration/.
- WebSphere MQ Integrator

  This software helps you to flexibly connect and integrate their assets within the enterprise and with trading partners. For more information, see http://www.ibm.com/software/integration/.

Conventions used in this guide

The guide uses several typeface conventions for special terms and actions. These conventions have the following meanings.

Bold	Lowercase and mixed-case commands, command options, and flags that appear within text appear like this, in **bold**. Graphical user interface elements, except for titles of windows and dialogs and names of keys also appear like this, in **bold**.
Italic	Variables, values you must provide, new terms, and words and phrases that are emphasized appear like this, in *italic* type.
`Monospace`	Commands, command options, and flags that appear on a separate line, code examples, output, and message text appear like this, in `monospace` type. Names of files and directories, text strings you must type, when they appear within text, names of Java methods and classes, and HTML and XML tags also appear like this, in `monospace` type.

The following variables are used in this documentation:

- *`$WVS_ROOT`* is a variable defined during the installation of WebSphere Voice Server. Typing `echo $WVS_ROOT` at a command prompt shows you the path where WebSphere Voice Server is installed, such as `. /opt/WebSphere/VoiceServer`.
- *`WAS_ROOT`* is a variable representing the installation path of WebSphere Application Server.

Chapter 2. Developing

This document concentrates on tasks associated with WebSphere Voice Server product usage and administration. There are many sources of excellent information on developing speech user interfaces and applications. The speech development environment provided with WebSphere Voice Server is the Voice Toolkit for WebSphere Studio, which can help streamline application design activities.

Although application development activities are typically not performed by the same professionals as voice server administration, use the sections below for an overview of the tasks involved in developing.

Standards for developing

This topic describes the supported standards for grammars and text-to-speech (TTS).

The following specifications are supported for developing grammars:

- Speech Recognition Grammar Specification (SRGS) 1.0 – http://www.w3.org/TR/speech-grammar/
- Semantic Interpretation Speech Recognition (SISR) 1.0 – http://www.w3.org/TR/semantic-interpretation/

WebSphere Voice Server TTS supports the following markup language.

- Speech Synthesis Markup Language (SSML) 1.0 – http://www.w3.org/TR/speech-synthesis/

Understanding VoiceXML

WebSphere Voice Server is an Media Resource Control Protocol (MRCP) version 1 draft 4 server. Thus, Voice Extensible Markup Language (VoiceXML) browsers that support the MRCP version 1 draft 4 standard can be used with WebSphere Voice Server. A VoiceXML browser is part of a comprehensive speech solution provided by a third party along with the gateway.

VoiceXML is the industry-standard markup language for interactive speech applications. With VoiceXML, you can use a Web-centric programming model and a markup language specially designed to bring the speed and flexibility of Web-based development and content delivery to interactive speech applications. In other words, development skills in HTML transfer to VoiceXML, improving the time to code speech applications.

VoiceXML dialogs provide a flexible presentation layer on top of the logic and data layers of your business applications. Using VoiceXML, you can build sophisticated dialogs that enable the caller to use either speech or key-presses to provide input, and to hear responses as either synthesized or prerecorded speech. The speech input from callers can be either recognized or recorded. VoiceXML enables you to build usable and flexible voice interfaces to your business.

In addition to accessing your business data, VoiceXML applications can also access telephony information such as called and calling number and perform simple call transfers. The recommended approach is to develop dialogs using VoiceXML in the complete solution. This can include host data accessed through enterprise Java beans (EJBs) and Java server programs (JSPs).

Developing a speech application

The development environment provided with the WebSphere Voice Server is the Voice Toolkit for WebSphere Studio.

The features of this programming environment are the following:

- Development environment (IDE)
- Call Flow Builder tool
- VoiceXML editor
- Call Control Extensible Markup Language (CCXML) editor
- Editors for creating grammars written in Speech Recognition Grammar Specification (SRGS) XML or ABNF
- Editors for testing grammars using speech, text, or enumeration

The toolkit also includes a pronunciation builder, an audio recorder, and a set of VoiceXML reusable dialog components.

For more information on the tools provided in Voice Toolkit for WebSphere Studio, see the Voice Toolkit information center on the Voice Toolkit CD. In order to access the Voice Toolkit information center, you must have the Voice Toolkit installed.

A comprehensive overview of speech application and interface design is provided in the *VoiceXML Programmers Guide*, which is a related document in the Voice Toolkit information center.

Testing grammars

When developing a speech application, you must also test your grammars. WebSphere Voice Server provides two different tools to test grammars – the Voice Toolkit for WebSphere Studio and the voiceTest tool.

The Voice Toolkit provides three GUI utilities for testing:

- A grammar test tool that interfaces with a remote WebSphere Voice Server
- Pronunciation migration tools for the new lexicon file format
- Lexicon file support for custom pronunciations

The command-line tool, voiceTest, also allows you to create a lexicon file using the -lexicon tag when defining a grammar. Both the Voice Toolkit and the voiceTest tool allow you to build lexicon files that can improve grammars.

For more information on the tools provided in Voice Toolkit for WebSphere Studio, see the Voice Toolkit information center on the Voice Toolkit CD. In order to access the Voice Toolkit information center, you must have the Voice Toolkit installed.

Testing Japanese grammars

After testing your Japanese grammar using the Voice Toolkit, you might find that the WebSphere Voice Server identifies certain Japanese words as ″missing″ and produces a grammar compilation error. Use the following steps to correct this error:

1. Open your grammar file using Voice Toolkit for WebSphere Studio.
2. Generate pool files from the grammar.
3. Convert the pool files to a lexicon file.

If the lexicon file is large, you should preload the grammar.

Tuning speech resources for effective applications

Tuning a speech application involves modifying the pronunciations of the WebSphere Voice Server automatic speech recognition (ASR) and text-to-speech (TTS) engines to recognize user input and pronounce system prompts more effectively.

This section includes tables showing the phonetic pronunciation (″sounds-like″ spelling) of vowels and consonants in each of the supported ASR languages.

WebSphere Voice Server run-time pronunciation errors are usually caused by the stress placed on the vowel sounds. If you experience pronunciation errors on certain words, modify the baseforms, to improve ASR. See Preparing a user dictionary file of pronunciations for instructions on how to do this. Use the tables in this section to assist you in correcting baseform phonetic sounds.

Preparing a user dictionary file of pronunciations

A user dictionary file enables you to define multiple pronunciations for words by specifying words and their associated pronunciations. Create this file using an ASCII text editor of your choice.

To be accessible to your application, the file must be named `dirname.dic` (where *dirname* is the name of the directory where the associated grammar files are located in the application’s output directory. The dictionary file must also be located in that directory).

The format of a word-baseform entry in a user dictionary file is: `word | baseform` where *word* is the spelling of word the engine should be able to recognize, and *baseform* is the phonetic pronunciation of the word.

For example, to define one pronunciation for the word ″evening″ and two alternative pronunciations for ″fifteen″, include the following lines in the user dictionary file:

```
evening | IY V N IX NG
fifteen | F IH F T IY N
fifteen | F IX F T IY N
```

See Tuning speech resources for effective applications for details of the character symbols used to specify phonetic pronunciations and the sounds they represent in each supported language.

Japanese phonetic pronunciation

The following table is a Japanese phonetic chart to assist you in correcting Japanese baseform phonetic sounds.

Table 1. Guide to the Japanese phonetic alphabet

Symbol	Description
Qb	sokuon+b
Qch	sokuon+ch
Qd	sokuon+d
Qf	sokuon+f
Qg	sokuon+g
Qh	sokuon+h
Qj	sokuon+j
Qk	sokuon+k
Qky	sokuon+ky
Qp	sokuon+p

Table 1. Guide to the Japanese phonetic alphabet (continued)

Symbol	Description
Qpy	sokuon+py
Qs	sokuon+s
Qsh	sokuon+sh
Qt	sokuon+t
Qts	sokuon+ts
Qz	sokuon+z
_n	hatsuon
a	vowel
a:	long vowel
b	ba gyou
by	youon
ch	cha gyou
d	da gyou
dy	dya gyou
e	vowel
e:	long vowel
f	fa gyou
g	ga gyou
gy	gya gyou
h	ha gyou
hy	hya gyou
i	vowel
i:	long vowel
j	ji
k	ka gyou
ky	kya gyou
m	nasal m
my	mya gyou
n	nasal n
ny	nya gyou
o	vowel
o:	long vowel
p	pa gyou
py	pya gyou
r	ra gyou
ry	rya gyou
s	sa gyou
sh	sha gyou
t	ta gyou
ts	tsu

Table 1. Guide to the Japanese phonetic alphabet (continued)

Symbol	Description
u	vowel
u:	long vowel
w	wa
y	youon
z	za gyou

Simplified Chinese phonetic pronunciation

The following table is a Simplified Chinese phonetic chart to assist you in correcting Simplified Chinese baseform phonetic sounds.

Table 2. Guide to the Simplified Chinese phonetic alphabet

Symbol	Description
“|” (pipe symbol)	Word boundary phone.
X	The silence phone at the end of each word.
D$	The silence phone at the beginning and end of a sentence.
B	“b” as in ”八”，”白”，”包”
BI	“bi” as in ”闭”，”表”，”比”
BU	“bu” as in ”不”，”步”，”波”
C	“c” as in ”草”，”擦”，”采”
CU	“cu” as in ”错”，”粗”，”村”
CH	“ch” as in ”吃”，”茶”，”车”
CHU	“chu” as in ”春”，”窗”，”吹”
D	“d” as in ”大”，”当”，”灯”
DI	“di” as in ”定”，”顶”，”地”
DU	“du” as in ”读”，”对”，”多”

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
F	“f” as in ″发″，″非″，″放″
G	“g” as in ″刚″，″盖″，″够″
GS	Zero initial phone as in ″安″，″爱″，″恩″
GU	“gu” as in ″工″，″国″，″关″
H	“h” as in ″海″，″河″，″寒″
HU	“hu” as in ″湖″，″火″，″会″
JI	“ji” as in ″江″，″机″，″进″
JU	“ju” as in ″军″，″捐″，″句″
K	“k” as in ″科″，″卡″，″抗″
KU	“ku” as in ″库″，″矿″，″阔″
L	“l” as in ″辣″，″来″，″楼″
LI	“li” as in ″力″，″灵″，″疗″
LU	“lu” as in ″路″，″乱″，″论″
LYU	“lü” as in ″绿″，″铝″，″律″
M	“m” as in ″马″，″猫″，″墨″
MI	“mi” as in ″米″，″名″，″秒″

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
MU	“mu” as in ″木″，″目″，″幕″
N	“n” as in ″拿″，″南″，″那″
NI	“ni” as in ″牛″，″宁″，″鸟″
NU	“nu” as in ″怒″，″暖″，″诺″
NYU	“nü” as in ″女″，″虐″
P	“p” as in ″盘″，″胖″，″旁″
PI	“pi” as in ″皮″，″平″，″品″
QI	“qi” as in ″期″，″桥″，″恰″
QU	“qu” as in ″全″，″去″，″确″
R	“r” as in ″日″，″让″，″燃″
RU	“ru” as in ″弱″，″如″，″软″
S	“s” as in ″丝″，″伞″，″洒″
SU	“su” as in ″松″，″算″，″岁″
SH	“sh” as in ″石″，″沙″，″射″
SHU	“shu” as in ″水″，″说″，″书″
T	“t” as in ″塔″，″特″，″汤″

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
TI	“ti” as in ″题″，″亭″，″体″
TU	“tu” as in ″同″，″拖″，″吞″
W	“w” as in ″挖″，″我″，″网″
XI	“xi” as in ″响″，″戏″，″夏″
XU	“xu” as in ″虚″，″讯″，″选″
Y	“y” as in ″羊″，″盐″，″影″
YU	“yu” as in ″云″，″远″，″月″
Z	“z” as in ″字″，″杂″，″责″
ZU	“zu” as in ″嘴″，″左″，″钻″
ZH	“zh” as in ″炸″，″站″，″照″
ZHU	“zhu” as in ″竹″，″追″，″桌″
A1	“a1” as in ″拉″，″他″，″八″
A2	“a2” as in ″拿″，″来″，″爬″
A3	“a3” as in ″马″，″把″，″洒″
A4	“a4” as in ″踏″，″罢″，″那″
A5	“a5” as in ″啦″

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
AI1	“ai1” as in ”拍”， ”胎”， ”该”
AI2	“ai2” as in ”白”， ”来”， ”爬”
AI3	“ai3” as in ”百”， ”凯”， ”买”
AI4	“ai4” as in ”败”， ”泰”， ”卖”
AI5	“ai5” as the last vowel in ”老太太” no tone.
AN1	“an1” as in ”山”， ”单”， ”甘”
AN2	“an2” as in ”环”， ”南”， ”圆”
AN3	“an3” as in ”感”， ”板”， ”点”
AN4	“an4” as in ”汗”， ”办”， ”慢”
AN5	“an5” as the last vowel in ”打算盘” no tone.
AG1	“ag1” as in ”光”， ”康”， ”乡”
AG2	“ag2” as in ”祥”， ”羊”， ”房”
AG3	“ag3” as in ”讲”， ”想”， ”场”
AG4	“ag4” as in ”唱”， ”上”， ”抗”
AG5	“ag5” as the last vowel in ”带上” no tone.
AO1	“ao1” as in ”猫”， ”涛”， ”包”

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
AO2	“ao2” as in ″劳″, ″条″, ″苗″
AO3	“ao3” as in ″恼″, ″好″, ″老″
AO4	“ao4” as in ″抱″, ″靠″, ″闹″
AO5	“ao5” as in ″奥″ no tone.
E1	“e1” as in ″车″, ″哥″, ″遮″
E2	“e2” as in ″蛇″, ″格″, ″咳″
E3	“e3” as in ″扯″, ″葛″, ″可″
E4	“e4” as in ″社″, ″个″, ″客″
E5	“e5” as in ″了″ no tone.
EI1	“ei1” as in ″杯″, ″归″, ″飞″
EI2	“ei2” as in ″回″, ″赔″, ″肥″
EI3	“ei3” as in ″北″, ″鬼″, ″匪″
EI4	“ei4” as in ″被″, ″翠″, ″费″
EI5	“ei5” as in ″呗″, ″嘞″ no tone.
EN1	“en1” as in ″闷″, ″喷″, ″分″
EN2	“en2” as in ″门″, ″盆″, ″焚″

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
EN3	“en3” as in ”本”, ”很”, ”粉”
EN4	“en4” as in ”笨”, ”趁”, ”份”
EN5	“en5” as in ”们” no tone.
EG1	“eg1” as in ”崩”, ”登”, ”称”
EG2	“eg2” as in ”甭”, ”棱”, ”成”
EG3	“eg3” as in ”耿”, ”冷”, ”等”
EG4	“eg4” as in ”蹦”, ”秤”, ”碰”
EG5	“eg5” as in ”哼” no tone.
ER1	“er1” as in Optional.
ER2	“er2” as in ”儿”
ER3	“er3” as in ”尔”
ER4	“er4” as in ”二”
ER5	“er5” as the last vowel in ”一块儿” no tone.
I1	“i1” as in ”逼”, ”低”, ”机”
I2	“i2” as in ”鼻”, ”敌”, ”及”
I3	“i3” as in ”比”, ”底”, ”几”

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
I4	“i4” as in ”必”，”地”，”计”
I5	“i5” as in ”璃”，”哩” no tone.
IE1	“ie1” as in ”接”，”街”，”切”
IE2	“ie2” as in ”别”，”节”，”协”
IE3	“ie3” as in ”撇”，”姐”，”且”
IE4	“ie4” as in ”月”，”界”，”列”
IE5	“ie5” as in ”咧” no tone.
IH1	“ih1” as in ”吃”，”疵”，”失”
IH2	“ih2” as in ”持”，”词”，”十”
IH3	“ih3” as in ”尺”，”此”，”使”
IH4	“ih4” as in ”赤”，”次”，”日”
IH5	“ih5” as the last vowel in ”铁尺” no tone.
IN1	“in1” as in ”彬”，”今”，”军”
IN2	“in2” as in ”林”，”民”，”您”
IN3	“in3” as in ”凛”，”紧”，”寝”

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
IN4	“in4” as in ”殡”, ”进”, ”俊”
IN5	“in5” as the last vowel in ”乡亲” no tone.
IG1	“ig1” as in ”拎”, ”丁”, ”经”
IG2	“ig2” as in ”零”, ”明”, ”宁”
IG3	“ig3” as in ”饼”, ”顶”, ”景”
IG4	“ig4” as in ”并”, ”定”, ”敬”
IG5	“ig5” as the last vowel in ”记性” no tone.
O1	“o1” as in ”玻”, ”多”, ”郭”
O2	“o2” as in ”伯”, ”夺”, ”国”
O3	“o3” as in ”跛”, ”躲”, ”果”
O4	“o4” as in ”播”, ”绰”, ”错”
O5	“o5” as in ”哟” no tone.
OG1	“og1” as in ”东”, ”通”, ”松”
OG2	“og2” as in ”穷”, ”同”, ”龙”
OG3	“og3” as in ”懂”, ”巩”, ”陇”

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
OG4	“og4” as in ”动”，”共”，”弄”
OG5	“og5” as the last vowel in ”演出合同” no tone.
OU1	“ou1” as in ”丢”，”抽”，”偷”
OU2	“ou2” as in ”谋”，”仇”，”头”
OU3	“ou3” as in ”剖”，”丑”，”否”
OU4	“ou4” as in ”凑”，”臭”，”漏”
OU5	“ou5” as in ”喽” no tone.
U1	“u1” as in ”屋”，”姑”，”忽”
U2	“u2” as in ”无”，”屠”，”胡”
U3	“u3” as in ”母”，”努”，”补”
U4	“u4” as in ”穆”，”怒”，”不”
U5	“u5” as the last vowel in ”子项目” no tone.
UN1	“un1” as in ”婚”，”春”，”吞”
UN2	“un2” as in ”魂”，”纯”，”屯”
UN3	“un3” as in ”盹”，”蠢”，”滚”

Table 2. Guide to the Simplified Chinese phonetic alphabet (continued)

Symbol	Description
UN4	“un4” as in ″盾″, ″混″, ″润″
UN5	“un5” as the last vowel in ″馄饨″ no tone.
YU1	“yu1” as in ″须″, ″淤″, ″居″
YU2	“yu2” as in ″徐″, ″鱼″, ″局″
YU3	“yu3” as in ″许″, ″与″, ″举″
YU4	“yu4” as in ″序″, ″育″, ″据″
YU5	“yu5” as he last vowel in ″饶有兴趣″ no tone.

UK English phonetic pronunciation

The following table is a UK English phonetic chart to assist you in correcting UK English baseform phonetic sounds.

Table 3. Guide to the UK English phonetic alphabet

Symbol	Description
“|” (pipe symbol)	Word Boundary Phone
AAS AAS	As in ARE fAther cAR
AES	As in At hAm trAp
AVS	As in Up bUt sOn
OAS OAS	As in Or fOUR Also
A_S ULS	As in OUt cOUnt hOW
AXS	As in tablE compAny commA
A_S ILS	As in EYE fIve lIE
_B.	As in But laBel caB
_CH	As in CHurch naTure reaCH
_D.	As in Dog meDal laD
_DF	As in THe boTHer baTHe
D$	Noisy silence, such as a keystroke or door slam
E_S	As in Any sAId bErry

Table 3. Guide to the UK English phonetic alphabet (continued)

Symbol	Description
E3S E3S	As in EArly wORd hER
E_S ILS	As in AId mAde grEY
_F.	As in First telePHone rouGH
_G.	As in Grab aGo biG
_H.	As in Home Hat beHind
ILS	As in Ink hYmn beIng
I_S I_S	As in EEl pEEp heavY
_JJ	As in Jay adJust chanGE
_K.	As in Cat baCKer maKe
_L.	As in Light vaLLey feeL
_M.	As in Mat haMMer hoMe
_N.	As in Night aNt baN
_NG	As in siNG beiNG loNG
AQS	As in hOt wAs Of
AXS ULS	As in gO knOW cOal
OAS ILS	As in OIl avOId toy
_P.	As in Park suPPle gaP
_R.	As in Right fRom thRee
_S.	As in Sit miSt maSS
_SH	As in SHarp naTIonal fiSH
_T.	As in To aTTend paT
_TH	As in Thing auThor paTH
ULS	As in gOOd shOUld pUt
U_S U_S	As in OOze bOOt whO
_V.	As in Very oVer saVe
_W.	As in White Was tWenty
X	Silence, such as a pause
_Y.	As in Yet Use bEAUty
_Z.	As in Zip siZe waS
_ZH	As in treaSure uSually viSion

US English phonetic pronunciation

The following table is a US English phonetic chart to assist you in correcting US English baseform phonetic sounds.

Table 4. Guide to the US English phonetic alphabet: vowels

Sound	Symbol	As in	Suggested	Comments
“ae”	AE	add cat pan	ae	The short “a” sound.

Table 4. Guide to the US English phonetic alphabet: vowels (continued)

Sound	Symbol	As in	Suggested	Comments
“ah”	AA	Are drop content	ah	Rarely wrong but sometimes an open ’o’ sound is used instead of the ’ah’. So spell ’mot’ as ’maht’ to change it from ’moat’ to ’mott’
“uh”	AH	One come	’Uh’ or double the following letter	Very rarely wrong; sometimes an unstressed sound is used instead. So spell ’Merom’ as ’Meromm’
Short “o”	AO	All off caution	or	Very rarely wrong. Try spelling ’Audi’ as ’Ordi’.
“au”	AW	How about	Ou au ow	If ’ow’ is mispronounced as ’oh’ try switching to ’au’ or ’ou’.
“eh”	AX	enable moment	eh	The unstressed “eh” sound as in “moment” or “enable” but **not** as in “enter.”
“er”	AXR	another over or	er	The unstressed “er” sound as in “another” over or “or” but **not** as in “her ”
Long “i”	AY	Fire why eye	Y ai add an e	Rare but if ’Byron’ is mispronounced more like “Birron” then try spelling it as ’Bairon’ or “Byeron” or even ’Byre-on’
“eh”	EH	Enter pleasant	eh	Rare but replacing the vowel letter or letters by ’eh’ usually works. For example if ’pleasant’ were mispronounced as “pleesant” then ’plehsant’ corrects it.
“er”	ER	Bird turn her	er	Never observed but ’er’ should work.
Long “a”	EY	Train eight A	Ai ay ey	If ’Wacome’ is mispronounced “Wah-come” then one of ’Waicome’ ’Waycome’ or ’Weycome’ should fix.
Short “i”	IH	If it pick	ih	The short I sound as in “it” “if ” or “pick” but **not** as in “lasting”. ’Piro’ should be pronounced with a short I and is mispronounced as “pyro” then ’pihro’ should work.
“ix”	IX	discuss saving budgeted decided	ix	The unstressed short “i” sound as in “discuss” “budgeted” “decided” or “saving”.
Long “e”	IY	Any he obvious	Ee ey ie	If ’Matrio’ was mispronounced as “mah-try-oh” then ’matreeo’ ’matreeoh’ or ’mahtreeoh’ should fix.

Table 4. Guide to the US English phonetic alphabet: vowels (continued)

Sound	Symbol	As in	Suggested	Comments
Long "o"	OW	Quote open go	Oh ow oe	'bow' should be right for 'Beau' and 'Dohl' should work for 'Dole'.
"oi"	OY	Avoid enjoy dive why	oy	The long "i" sound (or oi sound) as in "avoid" "employ" "dive" or "why".
Short "u"	UH	Good full could put	uh	The short "u" sound as in "good" "put" "full" or "could". If 'Clough' should be pronounced as in "cluf" then 'cluhf' should work.
Long "u"	UW	tooth blue view use you	Oo yoo	The stressed long "u" sound as in "to" "use" "you" or "view". Try 'Bloomehn' for the German pronunciation of 'Blumen'.

Usually nonvowel sounds will not be a problem for the Run Time baseform engine unless the name is of foreign extraction (as in the British pronunciation of Schedule (Schxhule). The table below lists the consonant sounds if you need to select a different sound than that produced by the automatic pronunciation rules.

Table 5. Guide to the US English phonetic alphabet: consonants

Sound	As in	Comments
"b"	be able	The "b" sound as in "be" or "able".
"bd"	sob tab blurb	The hard "b" sound at the end of a word.
"ch"	cheap child	The soft "c" sound before "h". The "ch" sound as in "cheap".
"d"	and David	The "D" sound as in "David" or "and".
"dd"		The "D" sound at the end of a word.
"dh"	the either	The voiced "th" sound as in "the" or "either" but NOT as in "thesis" or "thing".
"dx"	butter greater regarding order	The very short T or D sound as in "butter" "greater" "reagarding" or "order".
"d$"		A dummy sound that marks the beginning and end of a sentence.
"f"	father rough	The soft "f" sound.
"g"	again peg	The hard "g" sound.
"gd"	log tug	The G sound at the end of a word.
"hh"	here who	The "h" sound
"jh"	jeep roger	The soft "g" sound

Table 5. Guide to the US English phonetic alphabet: consonants (continued)

Sound	As in	Comments
“k”	because dark scale	The “k” sound but **not** the J sound at the beginning of a word.
“kd”	cake	The “k” sound at the end of a word.
“L”	level parallel	The “L” sound.
“m”	am must	The “m” sound.
“n”	final not none	The starting “N” or the unaccented N in the middle of a word as in “final” “not” or “none”.
“ng”	bang think singing	The “n” followed by a “g” or “k” sound.
“p”	adoption amps rapid	The “p” sound but **not** at the beginning of a word.
“pd”	pop tarp mop	The “p” sound at the end of a word.
“r”	abroad brace read	The “r” sound followed by a vowel sound in the same syllable.
“s”	sit circus decide	The “s” not followed by an “h” sound.
“sh”	action shade splash	The “sh” sound.
“t”	adapter retry let	The “t” (not “th”) sound not at the beginning of a word
“td”	date quit	The “t” sound at the end of word.
“th”	thesis thing	The unvoiced “th” sound but NOT as in “the” or “either”.
“ts”		The “ts” sound at the end of a word.
“v”	eleven improve very	The “v” sound.
“w”	frequent way question anywhere	The “w” sound.
“y”	emulate yes senior you	The “y” sound that leads into a vowel.
“z”	pans goes zero	The “z” sound.
“zh”	Asia pleasure Zho	The soft “z” sound.

Chapter 3. Planning

This section explains what you need to consider when you plan a speech system using WebSphere Voice Server. The requirements of the voice server system will vary depending on the following issues:

- The number of telephone lines or channels supported
- The call characteristics
- The speech application characteristics
- The language/voice installed
- The size and complexity of the grammars used (automatic speech recognition (ASR) only)

Planning your WebSphere Voice Server system involves the following steps:

1. Understanding the speech application.
2. Estimating speech resources for ASR and text-to-speech (TTS) engines.
3. Estimating resources for a LAN.
4. Identifying hardware and software requirements.
5. Creating a system topology: Single machine or multiple machines.

Understanding the speech application environment

In planning a voice system using WebSphere Voice Server speech technologies, you must determine:

- The number of gateway systems you require to connect to WebSphere Voice Server.
- The number of automatic speech recognition (ASR) engines and how many will be active at any one time.
- The number of concatenative text-to-speech (TTS) engines, and how many will be active at any one time.
- The number of machines you require, as a function of the number and speed of the processors available in each machine.
- The type of local area network necessary.

The number of licenses you require is calculated on the total number of processors on the machines on which the ASR and TTS engines are installed.

Note: It is not possible to give definitive information about exactly what size or number of machines you will need for your WebSphere Voice Server system in this documentation. Only approximate guidelines can be provided, and it is essential that any implementation be tested with realistic call volumes before it is put into production. For guidance about capacity planning for your specific configuration, contact your IBM representative.

Application load

To estimate the application load on the system, you need to know the following:

- The number of telephony channels required to handle the number of calls you expect for applications using WebSphere Voice Server technology. You must consider how long on average each caller is expected to wait before being answered.
- The call frequency and distribution – whether the number of calls likely to be made is the same throughout the day, or if the call pattern is likely to vary significantly, with peaks and troughs in demand.

Your WebSphere Voice Server system should be able to handle the maximum demand for speech resources. That is, the resources needed at the peak calling hour rather than a day's average number of hourly calls. The primary speech resource is the ASR or TTS engine. The demand for engines is influenced by both the frequency of calls and how they are distributed. If all the incoming calls use the same application and start at the same time, each call will need an engine at the same time so the demand will be high. If, on the other hand, calls are distributed normally, the number of engines needed simultaneously can be considerably smaller.

For your applications, you must determine the acceptable performance or desirability of an engine being available for a call without a significant delay. Delays can cause performance degradation (missed recognition or stuttering). If a degradation of performance is acceptable during peak utilization, fewer engines will be required.

- The time that engines are allocated (assigned) as a proportion of a call. This is known as the *allocation duty cycle*. ASR engines are allocated when a call is received and freed only when the call is disconnected. TTS engines are dynamically allocated on an as needed basis, that is, allocated to perform a single synthesis and then immediately freed.
- The time that engines are active (recognizing speech or synthesizing text to be played) as a proportion of a call. This is known as the *active duty cycle*. This can vary considerably, depending on the design of your voice applications and their complexity. These factors determine the number of concurrent ASR and TTS sessions that are required.

  The number of concurrent ASR and TTS sessions, in turn, determines the number of processors required and how powerful they must be. Similarly, these two variables – number and speed of the processors – dictate the number and size of the machines needed for your WebSphere Voice Server installation. As previously stated, the number of WebSphere Voice Server licenses you need to purchase is related to the number of processors and languages used.

  For example, a non-barge-in application using long prompts of synthesized text together with a simple grammar is likely to be actively engaged in recognition for only a short proportion of the length of a call. It will have a short active duty cycle for ASR but a long active duty cycle for speech synthesis.

  A barge-in application using shorter prompts of synthesized text together with a complex grammar is likely to spend more time actively engaged in recognition. In this case, the active duty cycle for ASR will be longer and the active duty cycles for speech synthesis shorter. If a system is underspecified, an engine might not be available at the start of a call.

The following figure shows a comparison of ASR and TTS engine allocation.

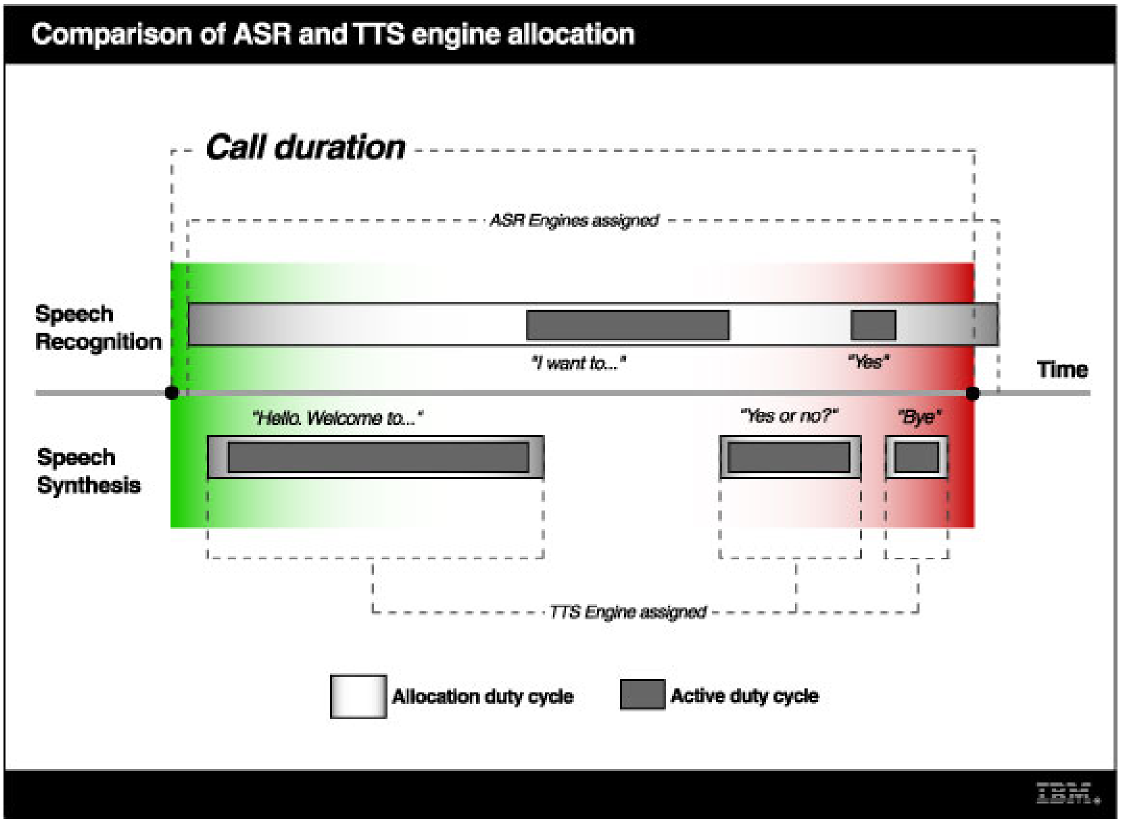


Figure 2. ASR and TTS sessions allocated for call duration

Application design

Some applications can be designed so that most callers are likely to complete their business and hang up quickly, thus freeing resources for another call. For more information about application design, refer to the Voice Toolkit information center provided when you install the Voice Toolkit for WebSphere Studio.

System resource

When you know the maximum number of concurrent sessions for ASR and/or TTS required, the recognition languages and voices required, and the complexity of your voice recognition applications, then you can determine how many WebSphere Voice Server machines are necessary, and their minimum specifications.

Remember: WebSphere Voice Server must run on its own machine.

A minimum base memory of 2GB is required on each WebSphere Voice Server recognition server machine. You can minimize the number of machines required by installing multiple high-speed processors and additional memory in each machine.

The actual number of WebSphere Voice Server engines that will run on each machine is solution-dependent. A solution must be tested to verify that a system can handle a condition where all of the WebSphere Voice Server engines are fully utilized. It also is important to ensure that the third-party gateway system can support all of the attached server machines and engines.

Related information

Estimating resources for ASR engines
Use this section to determine the resources needed for automatic speech recognition (ASR) engines.

Estimating resources for TTS engines
Use this section to determine the resources needed for text-to-speech (TTS) engines.

Estimating resources for ASR engines

Use this section to determine the resources needed for automatic speech recognition (ASR) engines.

The following factors influence the number of ASR engines you need to configure:

- The active duty cycle for CPU utilization (application load)
- The size and complexity of the grammar required by your voice applications
- The language installed

Note: The average CPU load should not exceed 80% for a machine. Doing so can increase the response time for results being passed back to the application. Another potential side effect is that utterances will not be recognized (ASR).

Estimating resources for TTS engines

Use this section to determine the resources needed for text-to-speech (TTS) engines.

The following factors influence the number of TTS engines you need to configure:

- The active duty cycle for CPU utilization (application load)
- The language installed
- The voice configured

During initialization of the WebSphere Voice Server, all voices for the installed language are loaded into a shared memory for use by any TTS engine. The voice(s) remain in memory until the WebSphere Voice Server is stopped. In addition, an output cache is created in shared memory to hold recently synthesized phrases. By default, the cache size is 20 MB, but it is configurable. The use of an output cache can improve synthesis processing, depending on an application‘s design and whether the same phrase is used repeatedly.

Note: The average CPU load should not exceed 80 percent for a machine. Doing so might increase the response time for results being passed back to the application. Another potential side effect is that synthesized text will stutter when played (TTS).

Estimating resources for a LAN

To handle the transport of voice data and control information between the client and the server, connect the WebSphere Voice Server to the third-party VoiceXML browser and gateway using a dedicated local area network (LAN). The minimum requirement is for a 100 Mbps Ethernet network, or equivalent. The LAN technology used must be appropriate for the number of channels and by extension, the number of automatic speech recognition (ASR) and text-to-speech (TTS) engines, configured for your installation.

It is difficult to be precise in estimating the saturation point for a specific configuration, because it is very sensitive to the type of external recognition system used and the nature of the voice application. If you have questions about these issues, contact your IBM representative who can give you advice on capacity planning.

Identifying hardware and software requirements

Hardware requirements

Supported hardware for WebSphere Voice Server include machines that meet the minimum hardware criteria defined below.

Table 6. Supported hardware

Processor	Memory (RAM)	Free Disk Space	Network	Other
Intel x86 1GHz or equivalent processor	2 GB	2 GB	TCP/IP	CD

Table 7. Text-to-speech (TTS) server additional disk space requirements

Language	WebSphere Voice Server
Japanese	417 MB
Simplified Chinese(For Simplified Chinese TTS, only a concatenative female voice is available.)	344 MB
UK English	593 MB
US English(The second US English female voice, Tyler, is 343 MB.)	612 MB

Software requirements

WebSphere Voice Server 5.1 supports the following operating systems:

- Red Hat Enterprise Linux WS/ES/AS for Intel 3.0 with Update 1 (2.4 Kernel)
- SUSE Linux Enterprise Server 8, powered by United Linux 1.0 (SLES-8) (Intel) with Service Pack 2a

See the WebSphere Voice Server product site for the latest information about supported operating systems.

Single machine topology

The maximum number of concurrent sessions a single machine can support is heavily dependent upon your hardware, the number and speed of the processors, and the amount of RAM on the machine. One machine does not provide any availability and failover support. If you take the server down for maintenance, your production environment is down. A single machine can be managed through the WebSphere Application Server administrative console providing ease of use for configuration, monitoring, tracing, alarming and logging.

The WebSphere Voice Server with the WebSphere Application Server base product runs on one machine. You can install the product in a standalone configuration or as part of a Network Dispatcher cluster in a multiple machine configuration. The standalone configuration is typically for developer desktops or standalone production computing, which involve a single Application Server instance operating independently of any other applications.

Different single machine topologies

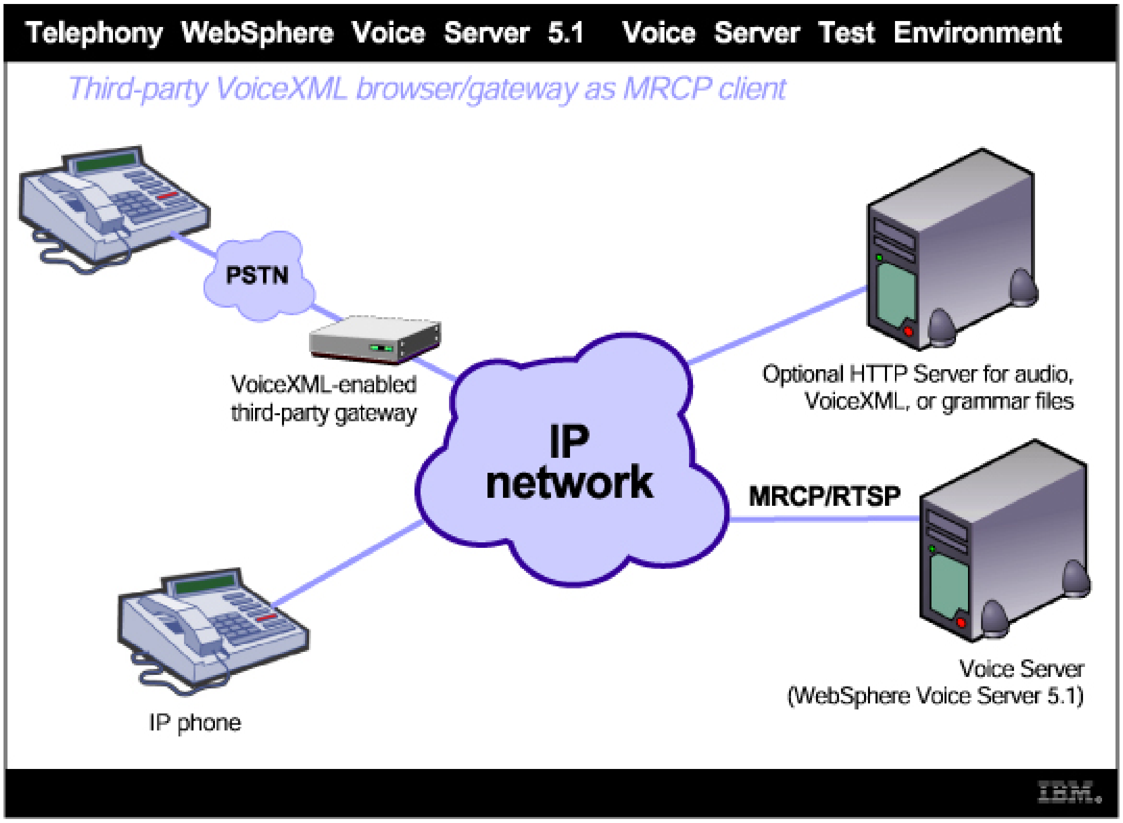
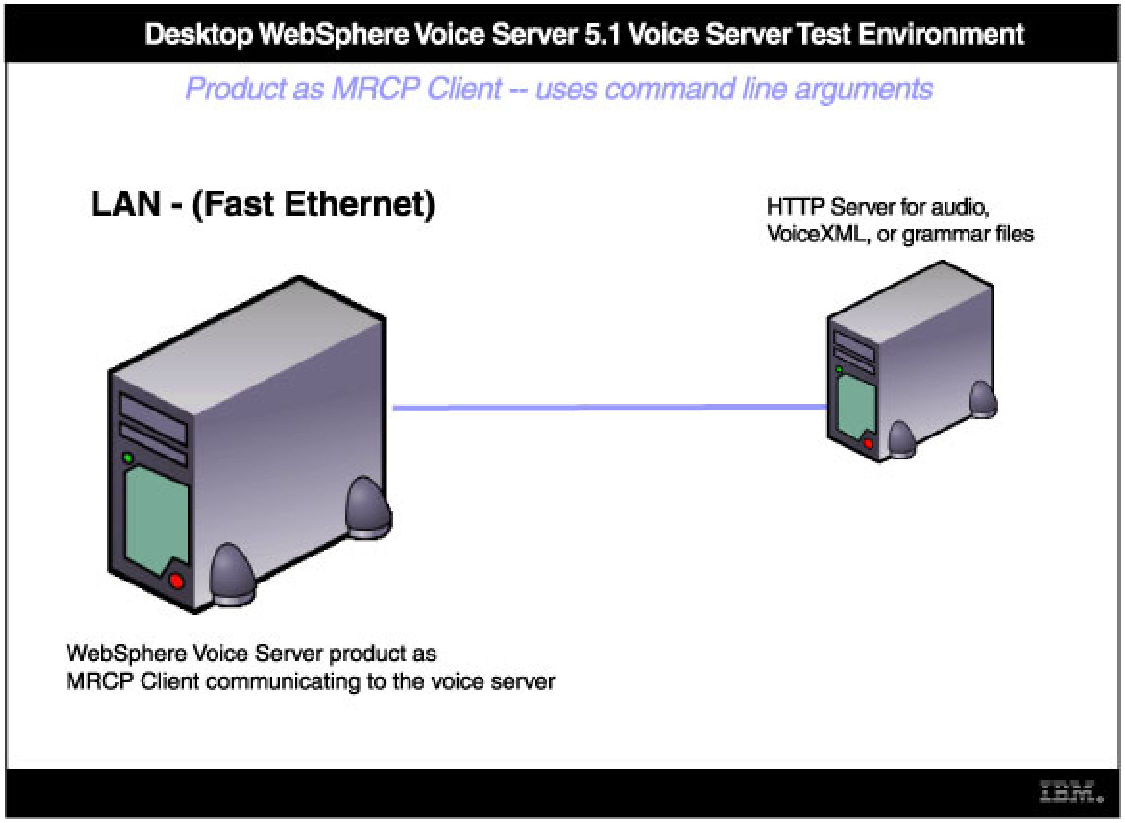

Telephony WebSphere Voice Server 5.1 Voice Server Test Environment
Third-party VoiceXML browser/gateway as MRCP client
PSTN
VoiceXML-enabled third-party gateway
IP network
Optional HTTP Server for audio, VoiceXML, or grammar files
MRCP/RTSP
Voice Server (WebSphere Voice Server 5.1)
IP phone
IBM.

Figure 3. The telephony environment

WebSphere Voice Server for Multiplatforms 5.1 supports Media Resource Control Protocol (MRCP) acting as an MRCP Server. MRCP is defined in the Internet Engineering Task Force (IETF) Internet Draft draft-shanmugham-mrcp-04. A VoiceXML browser that supports MRCP is considered an MRCP Client.

A third-party VoiceXML browser and gateway allows for telephony access to the Voice Server. If the VoiceXML browser runs inside a gateway, an HTTP Server might be required to provide the VoiceXML documents, audio, and grammar files because the gateway might be constrained regarding memory.



Desktop WebSphere Voice Server 5.1 Voice Server Test Environment

Product as MRCP Client -- uses command line arguments

LAN - (Fast Ethernet)
HTTP Server for audio, VoiceXML, or grammar files
WebSphere Voice Server product as MRCP Client communicating to the voice server

Figure 4. The desktop environment

For development purposes, a limited set of functionality is provided to test the WebSphere Voice Server without requiring a VoiceXML browser, gateway, and telephony infrastructure. This functionality is provided by the voiceTest program acting as an MRCP Client.

The voiceTest program enables:

- Testing a grammar against a prerecorded audio utterance and receiving the recognition result.
- Synthesizing a text document and saving the synthesized speech to an audio file. *The audio file can be played to hear the results*.
- Enhancing the lexicon documents referenced by a grammar.

When using the voiceTest program as the MRCP client, access to local audio and grammar files is provided by the file URI. An optional HTTP server can be used to provide the audio and grammar files, however, it must be installed on a separate machine (see the desktop environment figure above).

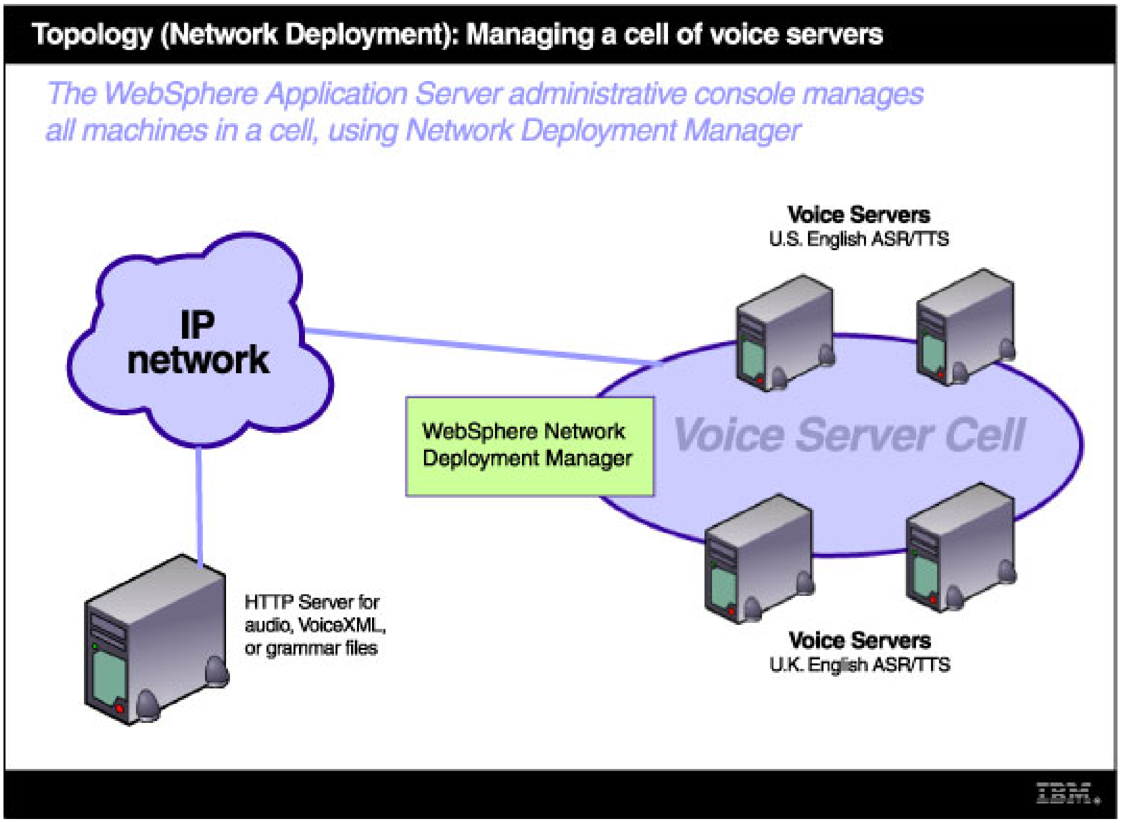
For more detailed development information, refer to the Voice Toolkit for WebSphere Studio information center.

Multiple machine topology

Multiple machine environments extend basic single-machine WebSphere Voice Server configurations by distributing the voice server solution over multiple machines, increasing the overall processing power from one machine to contributions from all machines in the configuration.

Multiple machine environments use two other WebSphere Application Server products included in the WebSphere Voice Server packaging:

- **Network Deployment** – Allows a single WebSphere Application Server administrative console to manage a cell of voice servers.
- **Network Dispatcher** – Allows for scalability, throughput, availability, failover, and maintainability.



Topology (Network Deployment): Managing a cell of voice servers

The WebSphere Application Server administrative console manages all machines in a cell, using Network Deployment Manager

Figure 5. Network Deployment topology

The preceding figure shows Network Deployment managing a cell of voice servers with different languages installed. An HTTP server is required to provide audio and grammar files to WebSphere Voice Server. Refer to the Administering section for more information.

Advantages of multiple machine topologies

Topologies that incorporate more than one cell have the following advantages:

- Isolation of hardware failure. If one cell goes offline due to hardware problems, the others can still process client requests.
- Isolation of software failure. Running a voice server in two or more cells isolates any problems that occur within a cell, while the other cells continue to handle client requests.

This isolation is helpful in a variety of situations:

- When rolling out a new voice server or a revision of an existing voice server.

  You can bring the new voice server or revision online in one cell, and test it in a live situation while other cells continue to handle client requests with the production version of the voice server.
- When deploying a new version of the WebSphere Voice Server and WebSphere Application Server software.

  You can bring the new version into production, and test it in a live situation without interrupting service.

- When applying fixes or patches to the WebSphere Voice Server and WebSphere Application Server software.

  You can take each cell offline to upgrade it, without interrupting the application.

If an unforeseen problem occurs with the new software, using multiple cells can prevent an outage to an entire site. You can also rollback to a previous software version more quickly. You can handle hardware and software upgrades on a cell-by-cell basis during off-peak hours.

Challenges of multiple machine topologies

Using multiple cells has a couple of challenges:
- Deployment is more complicated than for a single administrative cell.
- Multiple cells require more administration effort because each cell is administered independently. Use scripts to standardize and automate common administrative tasks to reduce this problem.

Establishing multiple machine environments

Perform the following steps to set up a Network Deployment environment on each of the systems that comprise the cell.

Steps for this task

1. Install WebSphere Voice Server (which installs the base WebSphere Application Server product) on each machine to create a WebSphere Voice Server node on the machine.

   The installation procedure is the same for a node that federates into a cell as it is for a standalone WebSphere Voice Server.

   There are several ways to federate a standalone Application Server installation instance into the deployment manager cell.
2. Install the WebSphere Application Server Network Deployment product on a machine. Only one system hosts the deployment manager. As the deployment manager federates base WebSphere Application Server nodes, it expands the cell that it manages. Although you can install a base WebSphere Application Server on the same machine as the deployment manager, it is not usual in a production environment unless you have a machine with the capacity to host both products. The deployment manager is the central administrative manager. It does not install the base WebSphere Application Server product on other machines. The only functions supported in the Network Deployment installation are the deployment manager and its associated administrative programs.
3. Start the deployment manager process.

   There are two ways to start the deployment manager:
   - As a monitored process, which restarts automatically if a failure occurs.
   - As an unmonitored process, which is what the startManager.sh or startManager.bat script does.

   Run the `startManager.sh` script from the `/bin` directory of the installation root of the deployment manager. For production systems, running the deployment manager as a monitored process is recommended.
4. Run the `addNode.sh` script on every node that you plan to federate into the cell. The `addNode.sh` script incorporates a base WebSphere Application Server product node into a deployment manager cell. You must run this tool on every system that you plan to make part of a Network Deployment cell. There are several parameters for the **addNode** command, but the most important are `includeapps`, the host name of the deployment manager node, the JMX connector type, and the JMX port of the deployment manager node, for example:

   ```
   addnode wasdoct 8889 -includeapps
   ```

   The example adds the base node on which the command runs to the cell managed by the `wasdoct` deployment manager node, using the default SOAP JMX connector type at port 8889. (Port 8889 is a coexistence value for the SOAP port. On the `wasdoct` development machine, the base node uses the default SOAP port.) The command includes all applications on the base node into the cell. As it federates the base node in response to the addNode script, the deployment manager also instantiates the node agent process, `nodeagent,` on the Application Server node.

Alternatively, you can use the administrative console of the deployment manager to add running Application Server nodes to the cell.

5. Enable the appropriate level of security after the installation is complete.
6. Install the WebSphere Voice Server Enterprise Archive (EAR) file.
7. Start all servers in the test environment.
8. Deploy the WebSphere Voice Server EAR in the test environment.
9. Run the installation verification test (IVT) for WebSphere Voice Server.
10. Prepare and monitor the environment into which you deploy applications.
11. Adjust application code, configurations, and system settings to improve performance.
12. Fix any known problems.
13. Set up your production system by configuring all server processes for monitoring by their operating systems.

What to do next

Now you can create a working WebSphere Voice Server and WebSphere Application Server cell.

Use the administrative console or other administrative tools to observe and control the incorporated nodes, and the resources on these nodes. The console provides a central location for configuring, monitoring, and controlling all Application Servers on all nodes within the cell.

Chapter 4. Installing

WebSphere Voice Server uses a graphical Launchpad and interface wizard that will guide you through the installation process. After installation, a First Steps interface provides a convenient place to explore the product. For WebSphere Voice Server version 5.1, the Launchpad and the installer wizard are in English.

Installing WebSphere Voice Server includes the following steps:

1. Determine type of installation: Single machine or multiple machines
2. Use the install wizard or install silently
3. Start the installation
4. Verify the installation
5. Explore WebSphere Voice Server
6. Uninstall

Installing WebSphere Voice Server

This topic introduces installing IBM WebSphere Voice Server, Version 5.1, including the tasks you need to perform.

Before you begin

To create a complete environment for installation, you must install any prerequisites and prepare the operating platform for installation. After installation, you can run the installation verification test and bring up WebSphere Voice Server.

What you need to know

Perform the following tasks to create a running version of the product on your machine.

Steps for this task

1. Plan to install an e-business network.

   This task helps you plan to install an e-business network. It also describes interoperability considerations.
2. Install the product.

   This task helps you prepare your machine for installation and explains the different types of installation available to you. It includes information on:
   - Using the LaunchPad
   - Using the silent installation method

   This task describes using the installation wizard. Through the installation wizard, you perform some initial configuration.
3. Verify the installation.
   a. Use the First Steps tool to run the Installation Verification Test.

      Launch the First Steps tool at the end of the installation.

   b. Identify and correct any problems using the troubleshooting procedure.
4. Set up a multinode environment.

   This task describes how to set up multiple nodes into a group or *cell* of Voice Servers, with centralized configuration under the control of the deployment manager node and its distributed node agents.

5. Automatically restart WebSphere Application Server processes.

   This task describes how to set up WebSphere Application Server processes for the operating system to monitor and restart. As default, WebSphere Voice Server starts when WebSphere Application server starts.

Results

The WebSphere Voice Server product is installed.

Uninstalling and reinstalling: See Uninstalling the product for information about uninstalling and reinstalling any WebSphere Voice Server product.

Related information

WebSphere Voice Server package for Version 5.1
WebSphere Voice server ships with a subset of WebSphere Application Server Network Deployment CDs.

firststeps command
The **firststeps** command starts the First Steps tool.

WebSphere Voice Server package for Version 5.1

WebSphere Voice server ships with a subset of WebSphere Application Server Network Deployment CDs.

Note: All WebSphere Application Server Network Deployment CDs are included for use by WebSphere Voice Server V5.1 only. The WebSphere Application Server base CD is not to be used on any machine other than the one on which WebSphere Voice Server is running. The WebSphere Application Server and included IBM HTTP Server are not to run any user applications or host user files. The WebSphere Application Server Network Dispatcher CD may be installed on a separate, single machine. The WebSphere Application Server Network Deployment CD may be installed on a separate, single machine as well.

The included WebSphere Application Server Network Deployment CDs are the following:

- **WebSphere Application Server CD for Linux**– The WebSphere Application Server product installation image contains the core application server runtime, IBM HTTP Server, and the node agent for communicating to the deployment manager when it is part of a cell.

  Note: The WebSphere Voice Server installation silently installs WebSphere Application Server.
- **Deployment Manager CD for Linux**– The Deployment Manager product installation image contains the deployment manager configured for use in departmental production computing scenarios. Although the Network Deployment package includes the base Application Server CD, installing the Network Deployment product does not install the base Application Server product. You must use the base product CD to install the base product.

  Note: Use the Deployment Manager installation on the CD after using the WebSphere Voice Server 5.1 base CD to install the voice system management functionality.
- **Edge Components CD for Linux**– The Edge components installation image contains IBM HTTP Server, and edge of network support for the Load Balancer (Dispatcher) and Caching Proxy (edge caching), as well as support for network authentication and single sign-on.

  Note: The Network Dispatcher only has the Edge Components installed. Use the installation provided on the Edge Components CD.

For more information about the complete WebSphere Application Server Network Deployment package, see the WebSphere Application Server Network Deployment 5.1 information center.

Related information

firststeps command
The **firststeps** command starts the First Steps tool.

firststeps command

The **firststeps** command starts the First Steps tool.

First Steps is a post-installation ease-of-use tool for directing WebSphere Voice Server elements from one place. Options dynamically appear on the First Steps panel, depending on features you install. With all options present, you can use First Steps to start and stop the server, verify the installation or access the information center.

First Steps must be started by the command line once the system has been rebooted, after the end of installation. If you try to verify the installation without rebooting, the system will not pass.

The location of the `firststeps.sh` script is:

- `$WVS_ROOT/firststeps/firststeps.sh`

Usage tips

You must select **Start the Server** before trying the installation verification test (IVT). The WebSphere Voice Server IVT does not automatically start the server.

Important: Do not select any other options from the First Steps panel after selecting the **Start the server** option until the server is started. There is a known problem if you select **Verify Installation** after selecting the **Start the server** option.

After selecting the **Start the server** option, an output screen displays with status messages. The success message informs you that the server is open for business. Then the menu item changes to **Stop the server**. If you are successful in starting the server, then proceed to verifying the installation if this is the first time you have used the First Steps panel since installing the product.

Click **Verify installation** to make sure that all is well with your installation.

Related information

WebSphere Voice Server package for Version 5.1
WebSphere Voice server ships with a subset of WebSphere Application Server Network Deployment CDs.

Installing the product

This topic describes how to install WebSphere Voice Server as the root user on a Linux platform.

The WebSphere Voice Server installation also installs WebSphere Application Server Network Deployment. If you create a CD copy of the product CD, create it as the root user on a Linux platform. This topic provides enough detail to guide you through preparing for, choosing, and installing the variety of options and features provided.

Before you begin

Read through this topic and its related topics to prepare for installation and to make yourself familiar with installation options, before you start to use the installation tools.

WebSphere Voice Server version 5.1 supports the same operating systems as WebSphere Application Server Network Deployment. WebSphere Application Server Network Deployment 5.1 (which includes Deployment Manager and Network Dispatcher) supports either of the following:

- Red Hat Enterprise Linux WS, ES, and AS 3.0 with Update 1 for Intel
- UnitedLinux 1.0 with Service Pack 2a or 3 for Intel

Read these topics before installing the product:
- Platform-specific tips for installing and migrating
- Using the Launchpad to start the installation
- Installing silently

As a general rule, if you launch an installation and there is a problem such as not having enough temporary space or not having prerequisite packages on your Linux system, cancel the installation, make the required changes, and restart the installation to pick up changes you made.

What you need to know

The Launchpad tool lets you access the product information center and the readme file. After starting the Launchpad, click the **Install the product** option to begin the installation wizard, which:
- Automatically checks prerequisites
- Asks the user for a full installation (which includes the WebSphere Application Server installation) or custom.

Steps for this task

1. Prepare your operating system platform for installation. For example, verify that you have enough disk space before starting installation.

   Tip: See Providing adequate disk space on Linux platforms for installation tips that can help you identify specific steps for you to take to prepare your operating system
2. Start the installation. The default installation method for all operating platforms is to click **Install the product** on the Launchpad tool to launch the Installation wizard, which is an InstallShield for MultiPlatforms (ISMP) application. This action launches the Installation wizard in English.

   The readme link in the Launchpad is to the `readme/readme_en.html` file on the base product CD.

   When you install, the current working directory must be the directory where the installer binary program is located. This placement is important because the resolution of the class files location is done in the current working directory.

   The following example shows how to change directory to the installation root before starting the installation on a Linux system:

   ```
   cd /$WVS_ROOT
    ./launchpad.sh
   ```

   The following example shows how to change directory to the CD when installing from the product CD on a Linux system:

   ```
   cd CD_mount_point
    ./launchpad.sh
   ```

   Failing to use the correct working directory can cause ISMP errors that stop the installation.

   You can start the Installation wizard from the product CD using the Launchpad (`launchpad.sh`). You can also start the installation using the **install.sh** command in the `dist` directory of the CD to bypass the Launchpad. This is required for silent installation.

   A silent installation uses the `-options responsefile` parameter, which causes the Installation wizard to read your responses from the options response file, instead of from the interactive user interface. You must customize the response file before installing silently. After customizing the file, you must issue the command to silently install. Silent installation is particularly useful if you install the product often. It requires install from the CD images on a local hard drive instead of install from the CDs.

   The rest of this procedure assumes that you are using the Installation wizard, launched from either `install.sh` or Launchpad. Corresponding entries in the response file exist for every prompt that is described as part of the wizard. Review the description of the response file for more information. Comments in the file describe how to customize its options.

3. Click **Next** to continue. The license agreement appears for you to read.
4. Click the radio button beside the **I accept the terms in the license agreement** message if you agree to the license agreement and click **Next** to continue. After you accept the licensing terms, the installation wizard checks for prerequisites and for previous versions.
5. Review the summary information and click **Next** to install the product code or **Back** to change your specifications.

   The summary panel displays the summary of choices for the installation, including target directory, disk space usage, and selected features.

   Note: WebSphere Voice Server only allows the selection of one language installation.

   When the installation is complete, the wizard displays the status of the installation.
6. Click **Finish** to close the installation wizard.

Results

You have now successfully installed WebSphere Voice Server and the features you selected.

The Installation wizard configures the product. You are not required to perform further configuration at this time. See What to do next for your next steps, including instructions on how to edit the startup profile.

What to do next

Examine the `/tmp/CWVOptInstall.log` file to verify that there were no file system or other unusual errors during installation. If there are problems, correct them, uninstall the product, and then reinstall. If there are no problems recorded in the `/tmp/CWVOptInstall.log`, verify or troubleshoot the installation in the next task described in Installing WebSphere Voice Server.

As a post-installation step prior to rebooting, you will also need to add *`wvs_install_path`*`/wvs.setenv` to your shell's startup profile (`.bash_profile` if you are using the bash shell on RedHat Linux). For example, edit the appropriate operating system level shell startup file and add the line:

```
. /opt/WebSphere/VoiceServer/wvs.setenv
```

Ensure that the startup profile is executed at boot up.

Uninstalling and reinstalling:

See Uninstalling the product for information about uninstalling WebSphere Voice Server.

Related information

Using the Launchpad to start the installation
Use the Launchpad to access a product overview, to access the readme file, and to install the product.

Installing silently
This topic describes how to perform a silent installation of the product as the root user on a Linux operating system.

Using the Launchpad to start the installation

Use the Launchpad to access a product overview, to access the readme file, and to install the product.

Steps for this task

1. Start the Launchpad.

   The Launchpad program is in the Linux root directory of the product CD.

You should start the Launchpad manually using a fully qualified command instead of changing directories to the CD and running the command locally from the platform directory.

Note: Mount the CD drive on Linux-based platforms. Consult the installation procedure for your platform for instructions on mounting CD drives. Open a shell window and issue a fully qualified **launchpad.sh** command to start the Launchpad, for example `mount/cdrom/launchpad.sh`.

The Launchpad opens in English only.

2. Use the Launchpad to access the product overview and the readme file.
3. Click **Install the product** to launch the installation wizard.

Results

You can use the Launchpad to start the installation.

You can launch the following links directly from your browser:

Product Overview
http://www.ibm.com/software/pervasive/voice_server/

It is possible that your security options might prevent redirection, such as to the http://www.ibm.com/software/info1/websphere/index.jsp?tab=products/voice Web site. If so, you can identify the IBM Web site as a trusted site to allow the redirection. For example, on Internet Explorer, add http://www.ibm.com to the list of Trusted Sites by selecting **Tools > Internet Options > Security (tab) > Trusted Sites**.

Readme File
`CD_mount_point/readme/readme_en.html`

Install the product
`CD_mount_point/dist/install.sh`

Related information

Installing silently
This topic describes how to perform a silent installation of the product as the root user on a Linux operating system.

Customizing the Network Deployment options response file
You can customize the response file to silently install the WebSphere Application Server Network Deployment product from a command line without the Installation wizard.

Installing silently

This topic describes how to perform a silent installation of the product as the root user on a Linux operating system.

What you need to know

Use this procedure to perform a silent installation of WebSphere Voice Server.

A silent installation uses the installation program to install the product. However, instead of displaying a wizard interface, the silent installation causes the installation program to read all of your responses from a file that you must customize. Silent installation requires install from the CD images on a local hard drive instead of install from the CDs.

Steps for this task

1. Verify that the user ID has root authority on Linux.
2. Copy the response file as **`myoptionsfile`** to your disk drive and customize it.

The name of the original file is wvs51response.txt, and it is located in the cdrom/responsefilesample directory.

3. On your local CD image, navigate to /dist/install.sh, and use the following command for silent installation:

   ```
   ./install.sh -silent -options full_path_to_response_file/responsefile.txt
   ```

   Note: The responsefile.txt is your customized copy of the wvs51response.txt, located in the cdrom/responsefilesample directory. The full path is required to the response file.

   The log.txt file is in the *$WVS_ROOT*/logs directory.

Results

You can install silently using the response file. Examine the log.txt file for lines similar to these:

```
(Jun 30, 2004 2:53:17 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction$OutputWatcher,
msg2, StdOut: ADMA6016I: Add to workspace adminconsole.war/WEB-INF/struts-config_1_0.dtd
(Jun 30, 2004 2:53:17 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction$OutputWatcher,
msg2, StdOut: ADMA6017I: Saved document
/opt/WebSphere/AppServer/wstemp/fd76e74534/workspace/cells/ngbuild1/applications/
adminconsole.ear/deployments/adminconsole/adminconsole.war/META-INF/MANIFEST.MF
(Jun 30, 2004 2:53:17 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction$OutputWatcher,
msg2, StdOut: ADMA6016I: Add to workspace adminconsole.war/META-INF/MANIFEST.MF
(Jun 30, 2004 2:53:17 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction$OutputWatcher,
msg2, StdOut: ADMA6017I: Saved document
/opt/WebSphere/AppServer/wstemp/fd76e74534/workspace/cells/ngbuild1/applications/
adminconsole.ear/deployments/adminconsole/adminconsole.war/WEB-INF/web.xml
(Jun 30, 2004 2:53:17 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction$OutputWatcher,
msg2, StdOut: ADMA6016I: Add to workspace adminconsole.war/WEB-INF/web.xml
(Jun 30, 2004 2:53:17 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction$OutputWatcher,
msg2, StdOut: ADMA5037I: Starting backup of app at
/opt/WebSphere/AppServer/wstemp/fd76e74534/workspace/cells/ngbuild1/applications/adminconsole.ear
(Jun 30, 2004 2:53:29 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction$OutputWatcher,
msg2, StdOut: ADMA5038I: Completed backup of app at
/opt/WebSphere/AppServer/wstemp/fd76e74534/workspace/cells/ngbuild1/applications/
adminconsole.ear/adminconsole.ear
(Jun 30, 2004 2:53:29 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction$OutputWatcher,
msg2, StdOut: ADMA6011I: Deleting directory tree /tmp/app_fd76e74937
(Jun 30, 2004 2:53:44 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction,
msg2, Return code = 0
(Jun 30, 2004 2:53:44 PM), CWVIN, com.ibm.wps.install.ExternalCommandAction,
msg2, Executing command: completed
(Jun 30, 2004 2:53:45 PM), CWVIN, com.ibm.wps.install.DeleteDirectoryOnExitAction,
msg1, Deleting directory tree on exit: /tmp/wvslaunch
```

This indicates that you have successfully installed the product.

Related information

Customizing the Network Deployment options response file
You can customize the response file to silently install the WebSphere Application Server Network Deployment product from a command line without the Installation wizard.

Installing WebSphere Voice Server console extensions on a deployment manager

Use the following steps to install WebSphere Application Server Network Deployment and the WebSphere Voice Server administrative console extensions to the deployment manager.

1. After installing WebSphere Voice Server (including base WebSphere Application Server), install WebSphere Application Server Network Deployment.

   When you install WebSphere Application Server Network Deployment, a deployment manager is created.

2. To add the WebSphere Voice Server administrative console extensions to the deployment manager, navigate to the console installation program (`install.sh`) located in the `admin` directory of the WebSphere Voice Server base product CD.

   Start the installation manually using a fully qualified command instead of changing directories to the CD and running the command locally from the platform directory.

   Note: Mount the CD drive on Linux-based platforms. Consult the installation procedure for your platform for instructions on mounting CD drives. Open a shell window and issue a fully qualified **`install.sh`** command to start the installation, for example `mount/cdrom/admin/install.sh`.

   The install opens in English only.
3. Click **Install the product** to launch the installation wizard.

Installing the WebSphere Edge Component: Load Balancer

Use the following steps to download and install Load Balancer:

1. To download the latest Edge Component, FTP to `service2.boulder.ibm.com/` from a command prompt. You can access this site with the userid `edgecust` and password `edge5fix`.
2. Navigate to the `edge/lb/5.1` directory and download `eLBLX-5.1.0.13.tar.gz`.
3. Use the instructions in the WebSphere Edge Server technote, Installation procedures for Load Balancer service releases, to get the license to install the `eLBLX-5.1.0.13.tar.gz` file.
4. Refer to the Load Balancer Administration Guide for detailed steps on installing and configuring the Load Balancer.
5. Update the iptables.

Customizing the options response file

You can customize the response file to silently install the WebSphere Application Server Network Deployment product from a command line without the Installation wizard.

For details on how to customize the response file, see Customizing the Network Deployment options response file.

Installation: Resources for learning

Use the WebSphere Application Server Network Deployment 5.1 information center to find relevant supplemental information about installation.

Installation tips

This topic describes tips specific to installing WebSphere Voice Server.

Refer to the following links for detailed tips for installation:

- Non-ASCII special characters are not supported in the name of the installation directory
- Verifying the system cp command when using emacs or other freeware

Using the installation verification test

After installing the product, modifying the startup profile, and rebooting your machine, you are ready to use the installation verification test (IVT).

What you need to know

The IVT program scans product log files for errors and verifies core functionality of the product installation.

Steps for this task

1. Start the server and wait for successful completion.
2. Once the server is running, select **Verify Installation** from the First Steps panel and observe the results in the First Steps status window. A log file for the installation is created and stored in the `logs` directory with the name of `ivt.log`. Do not select any other options from the First Steps panel after selecting **Verify Installation** and while the IVT is running. There is a known problem if you select the **Start the server** option at the same time that the IVT is running.
3. If the First Steps tool is not running, start it from the firststeps directory.

   `$WVS_ROOT/firststeps/firststeps.sh`

   The IVT program does not start the server automatically if the server is not running, so the server must be running before starting IVT. Once the server initializes, the IVT runs a series of verification tests and reports pass or fail status in a console window.

   The WebSphere Voice Server IVT program validates that your ASR language and TTS voices are installed correctly. Refer to the administrative console for the results of your installed languages and voices.

What to do next

Return to Installing WebSphere Voice Server to continue.

Uninstalling the product

This task describes how to uninstall WebSphere Voice Server and its related products.

Important: Before uninstalling, verify that you have no open Web browsers that are accessing the administrative console.

If WebSphere Voice Server, base WebSphere Application Server, and Network Deployment are all installed on the same machine, then you will need to uninstall them in the following order:

- WebSphere Application Server Network Deployment

  Note: If you do not have Network Deployment installed, refer to uninstalling base WebSphere Application Server for the recommended procedures for uninstalling unfederated base product nodes.
- WebSphere Application Server (base product)

  Note: If base WebSphere Application Server was installed by WebSphere Voice Server, then it will be uninstalled with WebSphere Voice Server.
- WebSphere Voice Server

To uninstall WebSphere Voice Server, use the following steps:

1. Change directory to `$WVS_ROOT/uninstall`.
2. Launch the uninstallation using the **./uninstall.sh** command.
3. Follow the screen options to uninstall WebSphere Voice Server, WebSphere Application Server, and your text-to-speech (TTS) and automatic speech recognition (ASR) engines.
4. Remove the `VoiceServer` directory.
5. Check for the entries listed in the vpd.properties file.

To uninstall the administrative console extensions for WebSphere Voice Server, use the following steps on the deployment manager:

1. Change directory to `$WVS_ROOT/uninstall`.
2. Launch the uninstallation using the **./uninstall.sh** command.
3. Follow the screen options to uninstall WebSphere Voice Server administrative console extensions.

4. Remove the `VoiceServer` directory.
5. Check for the entries listed in the vpd.properties file.
6. If necessary, uninstall WebSphere Application Server Deployment Manager.

Uninstalling Network Deployment

WebSphere Application Server Network Deployment provides an uninstaller program that you can use to uninstall the product. For details on how to uninstall Network Deployment, see Uninstalling Network Deployment in the WebSphere Application Server Network Deployment information center.

Uninstalling base WebSphere Application Server

For details on how to use the uninstaller program to uninstall WebSphere Application Server, see Uninstalling base WebSphere Application Server in the WebSphere Application Server Network Deployment information center.

Manually uninstalling on Linux platforms

This task describes uninstalling the product on Linux platforms.

Before you begin

Use the following procedure to remove all remnants of WebSphere Voice Server. This produces a clean system that allows you to reinstall the product. It is important to know the installation root directory for the product instance that you are removing.

Steps for this task

1. Use the **killall -9 java** command to kill all Java processes that are running.

   If there are Java processes running that are not related to WebSphere Application Server products, and it is not possible to stop them, stop all WebSphere Application Server product-related processes. Use the following command to determine all processes that are running:

   ```
   ps -ef | grep java
   ```

   Stop all WebSphere Application Server-related processes with the **kill -9** *java_pid_1 java_pid_2*... *java_pid_n* command.
2. Halt any running WebSphere MQ queue managers.
   a. Type **dspmq** to show the state of any queue managers.
   b. Type **endmqm -i** for each running queue manager.
   c. Type **$ ipcs -a** to check for any IPCs.
   d. Type **$ ipcrm -[qms] [ID]** to delete the IPCs.
3. Search for related packages. Type this command to search for related packages:
   - **rpm -qa | grep WS** to show packages for the WebSphere Application Server products and the IBM HTTP Server product

   Do not remove packages for WebSphere Application Server products that you are not uninstalling. You can identify packages that belong to version 5.1 by the 5.1 on the end of WSB packages and WSN packages.
4. If there are packages to delete, type **rpm -e** *packagename* to remove any WebSphere Application Server-related packages for the product you are uninstalling.

   You can remove more than one package with the following command:

   ```
   rpm -qa | grep
   package_prefix | xargs rpm -e
   ```

   However you should verify that everything in the list is something you want to remove before adding `| xargs rpm -e` to the command.

If there is a problem with package dependencies, you can use the following command to remove the packages:

```
rpm -e
packagename --nodeps --justdb
```

The nodeps option skips the dependency check. The justdb option updates only the package database, and not the file system. Using only the nodeps option can cause a failure in package removal if there is any mismatch in the dependent file system (files and directories).

5. Type **rm -rf /opt/WebSphere/AppServer/** to remove WebSphere Application Server directories in the instance installed in the `/opt/WebSphere/AppServer` installation root.
6. Type **rm -rf /opt/WebSphere/AppServer/ /opt/WebSphere/DeploymentManager/** to remove multiple directories. Again, do not remove directories for products that you intend to keep.
7. Type **rm -rf /opt/IBMHttpServer** to delete the IBM HTTP Server directory.

Results

When the process completes, the product is completely uninstalled. You are now ready to reinstall.

Usage scenario

Example of displaying package names beginning with wemps, for the embedded messaging feature

```
==>
rpm -qa | grep wemps
wemps-msg-De_DE-2.1.0-0
wemps-msg-Es_ES-2.1.0-0
wemps-msg-Fr_FR-2.1.0-0
wemps-msg-It_IT-2.1.0-0
wemps-msg-Ja_JP-2.1.0-0
wemps-msg-Ko_KR-2.1.0-0
wemps-msg-Pt_BR-2.1.0-0
wemps-msg-Zh_CN-2.1.0-0
wemps-msg-Zh_TW-2.1.0-0
wemps-runtime-2.1.0-0
```

Example of displaying package names beginning with WSN, for the WebSphere Application Server Network Deployment product

```
#
rpm -qa | grep WSN
WSNCO2AA51-5.1-0
WSNWG1AA51-5.1-0
WSNJD5AA51-1.3-1
WSNCF1AA51-5.1-0
WSNUR1AA51-5.1-0
WSNJD9AA51-1.3-1
WSNMQ1AA51-5.1-0
WSNCF4AA51-5.1-0
WSNMC5AA-5.0-0
WSNSR1AA51-5.1-0
WSNCO1AA51-5.1-0
```

What to do next

Return to Uninstalling the product to continue.

Federating multiple installation instances

You can federate or join a base WebSphere Application Server node to a WebSphere Application Server Network Deployment cell.

For detailed information on how to perform this task, see Federating multiple Version 5 installation instances.

Chapter 5. Configuring

This topic covers the configuration of voice servers.

After WebSphere Voice Server is installed, system administrators might want to modify several default settings. The **Configuring** topics in this documentation are geared towards your initial configuration and customization of the product, although you can adjust the product at any time. The product has been designed to require minimal setup, whenever possible. The primary interface for configuring WebSphere Voice Server is the administrative console.

Below are the configuration tasks for administrators of WebSphere Voice Server:

1. Configuring using the administrative console
2. Configuring using scripting and command line tools
3. Review all configuration parameters

Configuring using the administrative console

Launch and log in to the administrative console.

You can configure the automatic speech recognition (ASR) and text-to-speech (TTS) engines using the **Configuration** tab, which is displayed when you click on a voice server link on the Voice Servers panel.

From this view, you can access the ASR configuration parameters and the TTS configuration parameters.

For more information about the administrative console view, see Monitoring the voice server and resource status.

Configuring ASR settings

Use the following steps to configure automatic speech recognition (ASR) settings using the administrative console.

Remember: Configuration changes will only take effect after the voice server is stopped and restarted.

Specifying the number of ASR engines

1. Select which voice server you want to configure (for example, `dyn127-0-0-1_server1`), and click that voice server hyperlink in the Voice Servers panel.
2. Click the **Configuration** tab.
3. Navigate to the **Engines to start** parameter in the ASR configuration table. Place your cursor in the input box and type the desired number of engines to start. The number must be a positive integer. The maximum value is specific to your application and your hardware's capacity.
4. Click **Apply** to save your configuration changes and remain on the specific voice server view. Click **OK** to save your current configuration settings and return to the Voice Servers panel.

Preloading grammars

Preloading grammars improves the performance of a speech application, especially if the grammar is very large. Use the following steps to specify the location of a grammar file for preloading:

1. Select which voice server you want to configure (for example, `dyn127-0-0-1_server1`), and click that voice server hyperlink in the Voice Servers panel.
2. Click the **Configuration** tab.

3. Navigate to the **Grammar pre-loads** parameter in the ASR configuration table. Place your cursor in the input box and type the path and file name of the desired grammars. Multiple grammars can be separated by commas.

   Grammar examples:

   `http://vxidev1/pub/commands.grxml`

   or

   `http://vxidev1/pub/commands.grxml`, `http://vxidev1/pub/drinks.grxml`
4. Click **Apply** to save your configuration changes and remain on the specific voice server view. Click **OK** to save your current configuration settings and return to the Voice Servers panel.

Languages installed

The **Languages installed** parameter in the ASR configuration table is a read-only field that specifies the recognition language installed on your system. The language for ASR is determined at installation.

Configuring TTS settings

Use the following steps to configure text-to-speech (TTS) settings using the administrative console.

Remember: Configuration changes will only take effect after the voice server is stopped and restarted.

Specifying the number of TTS engines

1. Select which voice server you want to configure (for example, `dyn127-0-0-1_server1`), and click that voice server hyperlink in the Voice Servers panel.
2. Click the **Configuration** tab.
3. Navigate to the **Engines to start** parameter in the TTS configuration table. Place your cursor in the input box and type the desired number of engines to start. The number must be a positive integer. The maximum value is specific to your application and your hardware's capacity.
4. Click **Apply** to save your configuration changes and remain on the specific voice server view. Click **OK** to save your current configuration settings and return to the Voice Servers panel.

Preloading lexicons

Like preloading grammars, preloading lexicons also improves the performance of a speech application. Use the following steps to specify the location of a lexicon to preload:

Note: Lexicons for Japanese and Simplified Chinese are not supported by TTS.

1. Select which voice server you want to configure (for example, `dyn127-0-0-1_server1`), and click that voice server hyperlink in the Voice Servers panel.
2. Click the **Configuration** tab.
3. Navigate to the **Lexicon pre-loads** parameter in the TTS configuration table. Place your cursor in the input box and type the path and file name of the desired lexicon. Multiple lexicons can be separated by commas. There is no specific extension for lexicon files.

   Lexicon examples:

   `http://www.example.com/lexicon.file`

   or

   `http://www.example.com/lexicon.file`, `http://www.example.com/pronunciation.file`
4. Click **Apply** to save your configuration changes and remain on the specific voice server view. Click **OK** to save your current configuration settings and return to the Voice Servers panel.

Default voice

The **Default voice** is a modifiable parameter in the TTS configuration table that specifies which TTS voice to use as the MRCP server default voice. The current default voice is shown. This voice plays in your speech application whenever TTS is used and a voice is not specified. You can modify the **Default voice** by selecting a voice from the drop-down menu of installed voices.

Voice(s) installed

The **Voice(s) installed** parameter in the TTS configuration table is a read-only field that indicates the current TTS voice(s) and language installed on your system. TTS voices installed on your system are presented in a list, showing the language, gender, and voice name. The language for TTS is determined at installation.

Configuring using scripting and command line tools

Scripting

This documentation provides information on creating custom scripts for configuration.

These scripts only work on local servers and nodes. They cannot operate on a remote server or node. To administer a remote server, you can use the wsadmin tool connected to the Deployment Manager for the cell in which the target server or node is configured.

For more details about scripting, refer to the WebSphere Application Server Network Deployment 5.1 information center.

Command line tools

WebSphere Application Server provides command line tools for adding and removing voice servers from a voice server cell.

To manage using command line tools, perform the following steps:

1. Open a system command prompt.
2. Change to the `WAS_ROOT/bin` directory.
3. Run the desired command.

The command executes the requested function and displays the results on the screen. Refer to the command log file for additional information. When you use the -trace option for the command, the additional trace data is captured in the command log file. The directory location for the log files is under the default system log root directory, except for commands related to a specific server instance, in which case the log directory for that server is used. You can override the default location for the command log file using the -logfile option for the command.

For additional information, see Managing using command line tools in the WebSphere Application Server Network Deployment 5.1 information center.

Adding a voice server

This section gives instructions for adding a voice server using the command line interface.

The topology of a multiple-server WebSphere Voice Server deployment is a collection of voice servers (nodes). Each node has one application server instance, which has the WebSphere Voice Server Enterprise Archive (EAR) installed and active.

To learn more about the topology of WebSphere Voice Server, see the product overview.

Use the **addNode** command to add a voice server to a multiple-server WebSphere Voice Server deployment.

You must run this command from the `WAS_ROOT/bin` directory of a WebSphere Voice Server installation. Depending on the size and location of the new voice server you incorporate into the cell, this command can take a few minutes to complete.

Important: Be sure to back up your WebSphere Voice Server configuration before adding a voice server to a cell using the **backupVoiceServerConfig.sh** command-line utility.

Syntax

The command syntax is:

```
addNode <deploymgr host> <deploymgr port> -includeapps [options]
```

The `<deploymgr host>` and `<deploymgr port>` arguments are required. The default port number is 8879 for the default Simple Object Access Protocol (SOAP) port of the deployment manager. SOAP is the default Java Management Extensions (JMX) connector type for the command. If you install the deployment manager using coexistence ports, the default SOAP port is 8889. The **-includeapps** parameter is mandatory in a WebSphere Voice Server multiple machine environment.

Parameters

The following options are available for the **addNode** command:

-nowait
Tells the **addNode** command not to wait for successful initialization of the launched node agent process.

-quiet
Suppresses the progress information that the **addNode** command prints in normal mode.

-logfile <filename>
Specifies the location of the log file to which information gets written.

-trace Generates trace information into a file for debugging purposes.

-replacelog
By default, the **addNode** command appends to the existing trace file. This option causes the **addNode** command to overwrite the trace file.

-noagent
Tells the **addNode** command not to launch the node agent process for the new node.

-username <name>
Specifies the user name for authentication if security is enabled. Acts the same as the -user option.

-user <name>
Specifies the user name for authentication if security is enabled. Acts the same as the -username option.

-password <password>
Specifies the password for authentication if security is enabled.

-conntype <type>
Specifies the JMX connector type to use for connecting to the deployment manager. Valid types are SOAP or RMI, which stands for Remote Method Invocation.

-startingport <portNumber>
Supports the specification of a port number to use as the base port number for all node agent

and Java Messaging Service (JMS) server ports created during the **addNode** command. With this support you can control which ports are defined for these servers, rather than using the default port values. The starting port number is incremented to calculate the port number for every node agent port and JMS server port configured during the **addNode** command.

-help Prints a usage statement.

-? Prints a usage statement.

Usage scenario

The following examples demonstrate correct syntax:

```
addNode testhost 8879 -includeapps

./addNode.sh testhost 8879 -includeapps (from the remote node command line)

addNode deploymgr 8879 -includeapps -trace (produces the addNode.log file)

addNode host25 8879 -includeapps -nowait (does not wait for a node agent
process)
```

where

8879 is the default port.

Example output:

```
D:\WebSphere\AppServer\bin>addnode <dmgr_host> -includeapps
ADMU0116I: Tool information is being logged in file
           D:\WebSphere\AppServer\logs\addNode.log
ADMU0001I: Begin federation of node <node_name> with
deployment manager at
                                 <dmgr_host>:8879.
ADMU0009I: Successfully connected to deployment manager server:
<dmgr_host>:8879.
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU2010I: Stopping all server processes for node <node_name>
ADMU0512I: Server server1 cannot be reached. It appears to be
stopped.
ADMU0024I: Deleting the old backup directory.
ADMU0015I: Backing up the original cell repository.
ADMU0012I: Creating node agent configuration for node:
<node_name>
ADMU0014I: Adding node <node_name> configuration to cell:
<cell_name>
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0018I: Launching node agent process for node: <node_name>
ADMU0020I: Reading configuration for node agent process: nodeagent
ADMU0022I: Node agent launched. Waiting for initialization status.
ADMU0030I: Node agent initialization completed successfully.
Process ID is:
           2340
ADMU0523I: Creating queue manager for node <node_name> on
server jmsserver
ADMU0525I: Details of queue manager creation can be seen in the
file:
           createMQ.<node_name>_jmsserver.log
ADMU9990I:
ADMU0300I: Congratulations! Your node <node_name> has been
successfully
           incorporated into the <cell_name> cell.
ADMU9990I:
ADMU0306I: Be aware:
ADMU0302I: Any cell-level documents from the stand-alone
```

```
<node_name>
configuration have not been migrated to the new cell.
ADMU0307I: You might want to:
ADMU0303I: Update the configuration on the <cell_name>
deployment manager
           with values from the old cell-level documents.
ADMU9990I:
ADMU0306I: Be aware:
ADMU0304I: Because the -includeapps option was not specified,
applications installed on
           the stand-alone node were not installed on the new cell.
ADMU0307I: You might want to:
ADMU0305I: Install applications onto the <cell_name>  cell
using the wsadmin
           $AdminApp command or the administrative console.
ADMU9990I:
ADMU0003I: Node <node_name> has been successfully federated.
```

Removing a voice server

This section gives instructions for removing a voice server using the command line interface.

The topology of a multiple-server WebSphere Voice Server deployment is a collection of voice servers (nodes). Each node has one application server instance, which has the WebSphere Voice Server Enterprise Archive (EAR) installed and active.

To learn more about the topology of WebSphere Voice Server, see the product overview.

Use the **removeNode** command to remove a voice server from a multiple-server WebSphere Voice Server deployment.

You must run this command from the `WAS_ROOT/bin` directory of a WebSphere Application Server installation.

The **removeNode** command only removes the node-specific configuration from the cell. This command does not uninstall any applications that were installed as the result of executing an **addNode** command. Such applications can subsequently deploy on additional servers in the Network Deployment cell. As a consequence, an **addNode** command with the -includeapps option executed after a **removeNode** command does not move the applications into the cell because they already exist from the first **addNode** command. The resulting application servers added on the node do not contain any applications. To deal with this situation, add the node and use the deployment manager to manage the applications. Add the applications to the servers on the node after it is incorporated into the cell.

Depending on the size and location of the new node you remove from the cell, this command can take a few minutes to complete.

Syntax

The command syntax is:

```
removeNode [options]
```

All arguments are optional.

Parameters

The following options are available for the **removeNode** command:

-quiet
: Suppresses the progress information that the **removeNode** command prints in normal mode.

-logfile <fileName>
Specifies the location of the log file to which information is written.

-replacelog
Replaces the log file instead of appending to the current log.

-trace Generates trace information into a file for debugging purposes.

-statusport <portNumber>
Specifies that an administrator can set the port number for the node agent status callback.

-username <name>
Specifies the user name for authentication if security is enabled. Acts the same as the -user option.

-user <name>
Specifies the user name for authentication if security is enabled. Acts the same as the -username option.

-password <password>
Specifies the password for authentication if security is enabled.

-force Cleans up the local node configuration regardless of whether you can reach the deployment manager for cell repository cleanup. After using the `-force` parameter, you can need to use the **cleanupNode** command on the deployment manager.

-help Prints a usage statement.

-? Prints a usage statement.

Usage scenario

The following examples demonstrate correct syntax:

```
removeNode -quiet

removeNode -trace (produces the removeNode.log file)
```

Creating custom scripts for configurations

Configuration scripts use the WVSAdminConfig object to access the repository where configuration information is stored.

You can use the WVSAdminConfig object to list configuration objects and their attributes, create configuration objects, modify configuration objects, remove configuration objects, and obtain help:

1. Decide how you want to execute the script.

   If you want to run the script immediately from the command line, enter it surrounded by quotes as a parameter to the **wsadmin -c** command. To save the script for repeated use, compose it in a file and execute it with the **wsadmin -f** command. If you want to compose and run the script interactively, issue the **wsadmin** command without the -c or -f flags.

   For more information about executing scripts, see the WebSphere Application Server Network Deployment 5.1 information center.
2. Change to the *WAS_ROOT*/bin directory.
3. Write an WVSAdminConfig script command statement to update the number of automatic speech recognition (ASR) engines, for example:

   Using Jacl:

   ```
   set tmpCfg [$AdminControl queryNames node=ibm001,type=WVSAdminConfig,*]
   $AdminControl invoke $tmpCfg get com.ibm.voice.server.recognizer.asr.en_US.pool
   $AdminControl invoke $tmpCfg put {com.ibm.voice.server.recognizer.asr.en_US.pool 2 true}
   ```
4. Or write an WVSAdminConfig script command statement to set TTS pre-lexicon load values, for example:

Using Jacl:

```
set tmpCfg [$AdminControl queryNames node=ibm001,type=WVSAdminConfig,*]
$AdminControl invoke $tmpCfg get com.ibm.voice.server.synth.engine.preload-lexicons
$AdminControl invoke $tmpCfg put {com.ibm.voice.server.synth.engine.preload-lexicons
file:///tmp/grammar/lexi1.lex false}
```

You should also review Modifying nested attributes with the wsadmin tool.

Configuration Service MBean

Use the WebSphere Voice Server Configuration Service managed bean (MBean), WVSAdminConfig, to invoke configuration commands and to create or change elements of the WebSphere Voice Server configuration, such as the text-to-speech (TTS) or automatic speech recognition (ASR) engine configurations.

You can start the scripting client without a running server, if you only want to use local operations. To run in local mode, use the `-conntype NONE` option to start the scripting client. You will receive a message that you are running in the local mode. If a server is currently running it is not recommended to run WVSAdminConfig in local mode.

The following commands are available for the WVSAdminConfig MBean:

getlist
Returns a list of all the configuration technical names matching a filter string.

Note: The wildcard (*) attribute does not have to be after a period. For example, `com.ibm.voice.server.asr*`.

get Returns the value to which the specified technical name is mapped in the configuration.

put Updates the value for the technical name. Neither the technical name nor the value can be null.

Remember: Configuration changes will only take effect after the voice server is stopped and restarted.

Sample code

The following sample shows you how to find and access the WVSAdminConfig MBean interface and services.

```
wsadmin> set wvscfg [$AdminControl queryNames *:*,type=WVSAdminConfig]
WebSphere:cell=nextgen,name=WVSAdminConfig,
mbeanIdentifier=cells/nextgen/nodes/nextgen/servers/server1/serverl.xml#CustomService_1079378808193,
type=WVSAdminConfig,node=nextgen,process=server1

wsadmin> $Help operations $wvscfg
Operation
java.util.Properties getList(java.lang.String)
java.lang.String get(java.lang.String)
void put(java.lang.String, java.lang.String)
void put(java.lang.String, java.lang.String, boolean)
void put(java.util.Properties, boolean)
void backup(java.lang.String)
void restore(java.lang.String)
wsadmin> $AdminControl invoke $wvscfg get com.ibm.voice.server.rr.fetchthreads
11
wsadmin> $AdminControl invoke $wvscfg getList *
{com.ibm.voice.server.synth.mrcp.Zhou-Hong.speech-language zh_CN}
{com.ibm.voice.server.synth.mrcp.Shizuka.voice-age adult}
{com.ibm.voice.server.synth.mrcp.Allison.voice-age adult}
{com.ibm.voice.server.synth.mrcp.Shizuka.prosody-range 10}
{com.ibm.voice.server.synth.mrcp.Daichi.prosody-rate 50}
{com.ibm.voice.server.synth.mrcp.Shizuka.speech-language ja_JP}
{com.ibm.voice.server.synth.mrcp.Shizuka.prosody-pitch 268}
```

```
{com.ibm.voice.server.synth.mrcp.Andrew.prosody-rate 50}
{com.ibm.voice.server.synth.mrcp.Zhou-Hong.prosody-range 10}
{com.ibm.voice.server.rr.cachefilesize 9}
{com.ibm.voice.server.synth.mrcp.Shizuka.prosody-rate 50}
{com.ibm.voice.server.synth.mrcp.Zhou-Hong.prosody-pitch 137}
{com.ibm.voice.server.rr.caching true}
{com.ibm.voice.server.synth.mrcp.Daichi.voice-age adult}
{com.ibm.voice.server.synth.mrcp.David.voice-age adult}
{com.ibm.voice.server.recognizer.dtmf.pool 0}
{com.ibm.voice.server.synth.mrcp.Zhou-Hong.prosody-rate 50}
{com.ibm.voice.server.synth.mrcp.Andrew.voice-age adult}
. . .
wsadmin> $AdminControl invoke $wvscfg put {com.ibm.voice.server.rr.fetchthreads 20}
wsadmin> $AdminControl invoke $wvscfg put {com.ibm.voice.server.rr.fetchthreads 20 true}
wsadmin> $AdminControl invoke $wvscfg putProperties {{{com.ibm.voice.server.rr.fetchthreads 20}
{com.ibm.voice.server.synth.pool-size 3} {com.ibm.voice.server.common.codec G711ulaw} } false}
```

Configuration parameters

This section lists configuration parameters by categories of parameters related to the cache, automatic speech recognition (ASR) configuration, text-to-speech (TTS) configuration, and Real Time Streaming Protocol (RTSP) Bridge resources.

Cache-related parameters

WebSphere Voice Server has two different levels of caches. It has a resource first level cache available to hold grammars, lexicon documents, and audio files. It has a grammar second level cache available to hold the compiled grammars. These two levels of caches allow for the quick loading of grammars that are shared between ASR engines.

The following parameters control the two levels of caching. They identify the size and location of the caches. If you are developing your application, it is highly recommended to disable the first level cache until your application errors have been fixed.

Table 8. Cache-related parameters

Name	Technical Name	Description	Default value	Valid values
Grammar (second level) Default Cache Size (MB)	com.ibm.voice.server.gm.cachefilesize	Default grammar file cache capacity	9 MB	A positive integer value in mega bytes
Grammar (second level) Cache Location	com.ibm.voice.server.gm.cachename	Default location of where the Compiled Grammars are stored on the local file	*$WVS_ROOT*/gmcache	Must be a valid US-ASCII directory on the local file system (optionally ending in the file separator character such as /)
General (first level) Default Expiration Time	com.ibm.voice.server.rr.fetchexpires	Default value for all resources if expiration is not provided by the fetched resource	86400000 ms	A positive integer value in milliseconds

Table 8. Cache-related parameters (continued)

Name	Technical Name	Description	Default value	Valid values
General (first level) Cache Location	com.ibm.voice.server.rr. cachename	Location of the first level general cache	`$WVS_ROOT/rrcache`	Must be a valid US-ASCII directory on the local file system (optionally ending in the file separator character such as /)
General (first level) Memory Cache Size	com.ibm.voice.server.rr. cachememsize	General first level memory cache capacity	2 MB	A positive integer value in mega bytes
General (first level) Caching Enabled	com.ibm.voice.server.rr. caching	Enable/disable caching	True	True/False

ASR configuration parameters

The following parameters control the ASR engine. These parameters are also present in the ASR configuration table in the administrative console. For more information on the parameters from the console view, see Configuring ASR settings

Table 9. ASR configuration parameters

Name	Technical Name	Description	Default value	Valid values
Engines to start **Note:** The value for this parameter matches the number of **Engines to start** on the administrative console ASR configuration table.	com.ibm.voice.server.recognizer. asr.en_US.pool com.ibm.voice.server.recognizer. asr.en_GB.pool com.ibm.voice.server.recognizer. asr.ja_JP.pool com.ibm.voice.server.recognizer. asr.zh_CN.pool • en_US is US English • en_GB is UK English • ja_JP is Japanese • zh_CN is Simplified Chinese	Number of ASR engines to be started for each language. Zero (0) means that no ASR support for that language is configured.	0	0 to the maximum number of engines that can be hosted by your hardware configuration

Table 9. ASR configuration parameters (continued)

Name	Technical Name	Description	Default value	Valid values
Grammar preloads **Note:** The value for this parameter matches the number of **Grammar pre-loads** on the administrative console ASR configuration table.	com.ibm.voice.server.recognizer.grammar.preload	Defines a list of grammars to be preloaded when the system is started. If the URI represents the source of the grammar, it will be compiled before it is loaded.	Empty	List of URIs separated by commas

TTS configuration parameters

The following parameters control the TTS voice(s) installed on your Voice Server.

The gender and name of TTS voices are not modifiable and are as follows:

Japanese

Daichi – adult male

Shizuka – adult female

Simplified Chinese

Zhou-Hong – adult female

U.K. English

David – adult male

Dawn – adult female

U.S. English

Andrew– adult male

Allison and Tyler – adult female voices

The configurable (unless otherwise indicated) parameters for these voices are listed below.

Table 10. TTS voice configuration parameters

Name	Technical Name	Description	Default value	Valid values

Pitch	com.ibm.voice.server.synth.mrcp.*voicename*.prosody-pitch	Speaker's base pitch in hertz (Hz)	Andrew = 109Hz
			Allison = 226Hz
			Tyler = 226Hz
			David = 106Hz
			Dawn = 198Hz
			Zhou-Hong = 115Hz
			Daichi = 101Hz
			Shizuka = 268Hz
				40 - 422Hz

Pitch range	com.ibm.voice.server.synth.mrcp.*voicename*.prosody-range	Speaker's pitch fluctuation	Andrew = 126Hz
			Allison = 190Hz
			Tyler = 186Hz
			David = 126Hz
			Dawn = 158Hz
			Zhou-Hong = 150Hz
			Daichi = 95Hz
			Shizuka = 163Hz
				[0-9]+"Hz"

Loudness	com.ibm.voice.server.synth.mrcp.*voicename*.prosody-volume	Loudness	92Hz	0 - 100Hz

Rate	com.ibm.voice.server.synth.mrcp.*voicename*.prosody-rate	Speaker's speaking rate in words per minute (wpm)	Andrew = 184
			Allison = 186
			Tyler = 186
			David = 155
			Dawn = 150
			Zhou-Hong = 240
			Daichi = 118
			Shizuka = 144
				70 - 1297

Additional parameters for the system's voices are the ones that map to the TTS configuration table in the administrative console. For more information on the parameters from the console view, see Configuring TTS settings

Table 11. Additional TTS voice configuration parameters

Name	Technical Name	Description	Default value	Valid values
Lexicon preload	com.ibm.voice.server.synth.engine.preload-lexicons	List of lexicon files to be pre-loaded for all TTS engines	Empty	List of URIs separated by commas
Engines to start **Note:** The value for this parameter matches the number of **Engines to start** on the administrative console TTS configuration table.	com.ibm.voice.server.synth.pool-size	Number of TTS engines to be started.	0	The number must be a positive integer. The maximum value is specific to your application and your hardware's capacity.
Default voice **Note:** The value for this parameter matches the number of **Default voice** on the administrative console TTS configuration table.	com.ibm.voice.server.synth.default-voice	Default voice name used by all TTS engines	For Japanese, Daichi. For Simplified Chinese, Zhou-Hong. For UK English, David. For US English, Andrew.	String

RTSP parameters

These parameters define the port and URL to use when contacting the WebSphere Voice Server from the third-party VoiceXML 2.0 Browser (such as the MRCP Client).

Table 12. RTSP parameters

Name	Technical Name	Description	Default value	Valid values
RTSP Port	com.ibm.voice.server.rtspbridge.port	This is the port from which the RTSP server listens.	554	Any available port
RTSP Recognizer URL	com.ibm.voice.server.rtspbridge.recourl	RTSP media resource URL for the recognizer	/media/recognizer	Any valid relative URL
RTSP Synthisizer URL	com.ibm.voice.server.rtspbridge.synthurl	RTSP media resource URL for the synthesizer	/media/synthesizer	Any valid relative URL

Table 12. RTSP parameters (continued)

Name	Technical Name	Description	Default value	Valid values
RTSP Teardown	com.ibm.voice.server.rtspbridge.oneteardown	Allows the server to accept one teardown from the client in order to clean up all resources. If you set the value to true, the server will only expect one RTSP Teardown. However, if you set it to false (the default), then the server will expect an RTSP Teardown for both ASR and TTS.	False	True/False
RTSP Socket Time Out	com.ibm.voice.server.rtspbridge.sockettimeout	The amount of time RTSP Bridge will wait for a request from the client before timing out while in an established session	300	A positive integer value in seconds
Default Audio Codec Type	com.ibm.voice.server.common.codec	Type of audio codec	G711ulaw	G711ulaw or G711alaw

Chapter 6. Administering

Administration is the set of tasks by which you modify, control, and monitor your voice server system for general use. The primary interface for system administration is the administrative console. However, a command line interface also provides these capabilities.

Use the steps below in administering WebSphere Voice Server:

1. Using the administrative console
   - Filtering on the administrative console
   - Enabling security
   - Monitoring voice servers and resource status
   - Starting voice servers
   - Stopping voice servers
2. Managing using scripting

Using the administrative console

What you need to know

The administrative console is a graphical interface for performing deployment and system administration tasks. It runs in your Web browser.

You can use the console to perform tasks such as:

- Start and stop voice servers
- Enable product security, including access to the administrative console
- Monitor voice server and resource status

Starting and stopping the administrative console helps you begin using the console so that you can explore the available options.

Steps for this task

1. Start the server for the administrative console for a single machine topology, or start the Network Deployment manager process for multiple machine topologies.
2. Access the administrative console. After you point a Web browser at the URL for the administrative console, enter your user ID and, if needed, a password on a Login page.
3. Browse the administrative console. The important interfaces for general-purpose voice server administration are accessed using the Voice Servers panel.
4. Specify console preferences.

Starting and stopping the administrative console

What you need to know

To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

Steps for this task

1. Start the administrative console.

a. Enable cookies in the Web browser that you use to access the administrative console in order for the administrative console to work correctly.

b. In the same Web browser, type `http://your_fully_qualified_server_name:9090/admin` where *your_fully_qualified_server_name* is the fully qualified host name for the machine that contains the administrative server.

   When the administrative console is on the local machine, *your_fully_qualified_server_name* can be `localhost`. If security is enabled, your request is redirected to `https://your_fully_qualified_server_name:9043/admin`.

c. Wait for the console to load into the browser.

If you cannot start the administrative console because the console port conflicts with a Web application already running on the same machine, you must change the port number used by administrative console in the following files:

```
WAS_ROOT/config/cells/cell_name/nodes/node_name/servers/server_name/server.xml
```

```
WAS_ROOT/config/cells/cell_name/virtualhosts.xml
```

Change all occurrences of port 9090 to the desired port for the console. Alternatively, shut down the other application that uses the conflicting port before starting the WebSphere Voice Server product.

2. A Login page appears after the console starts. Log into the console:

   a. Enter your user name (user ID).

      Changes made to server configurations are saved using the specified user ID. Temporary server configurations are saved to the user ID if there is a session timeout or if the user logs out without saving the configuration changes.

      A user ID lasts for the duration of the session for which it was used to log in. If you enter an ID that is already in use (and in session), you are prompted to do one of the following:

      - Force the existing user ID out of session. The configuration file used by the existing user ID is saved in the temporary area.
      - Wait for the existing user ID to log out or time out of the session.
      - Specify a different user ID.

   b. If the console is secure, you must also enter a password for the user name. The console is secure if the following steps have been performed:

      1) Specified security user IDs and passwords

      2) Enabled global security

   c. Click **OK**.

3. Stop the administrative console. Click **Save** on the console taskbar to save work that you have done and then click **Logout** to exit the console.

   Note that if you close the browser before saving your work, when you next log in under the same user ID, you can recover any unsaved changes.

What to do next

If you have difficulty using the administrative console on Linux, try using the Netscape Communicator 7.1 browser based on Mozilla 1.0. The browser release is not officially supported by the WebSphere Voice Server product, but users have been able to access the console successfully with it.

Login settings

Use this page to specify the user for the WebSphere Voice Server administrative console. If you are using global security, then you must also specify a password.

Specifying a user allows you to resume work done previously with the product. After you type in a user ID, click **OK** to proceed to the next page and access the administrative console.

To view this page, start the administrative console.

The following are fields and settings relevant to the login window.

User ID
Specifies an entry identifying the user. The user ID must be unique to the administrative server.

While logged in, any changes you make during the session are saved to a configuration identified by your user ID. The next time you log in to the console with your user ID, your work will be restored.

If you are logging in from a client machine that currently has a browser open on the administrative console and you connect to the same server, your two connections might share the same HTTP session object, whether or not you are logging in under different user IDs. You must use two different types of browsers (for example, Microsoft Internet Explorer and Netscape Communicator) to log in to the administrative console from the same client machine in order to work from two different HTTP session objects, whether or not you use the same user ID. That is, if you need two concurrent sessions on the same client machine, access the administrative console from two different browser types.

Note that you can use the same browser type to log in twice to the console, and use different user IDs. For example, suppose that Joe logs into the administrative console under the user ID *joe* on the physical host *myhost1* using an Internet Explorer browser. Joe can log into the same console a second time using an Internet Explorer browser, but he must use a different user ID (such as *peggy*).

Another user is currently logged in with the same user name
Specifies whether to log out the user and continue work with the user ID specified or to return to the Login page and specify a different user ID or wait for the user to log out.

This field appears if the user closed an administrative console browser window without logging out, then opened a new browser and tried accessing the administrative console, or if there is already another user logged in under the same name.

Work with the master configuration
When enabled, this setting specifies that you want to use the default administrative configuration and not use the configuration last used for the user's session. Changes made to the user's session since the last saving of the administrative configuration will be lost.

This field appears only if the user changed the administrative configuration and then logged out or closed the Web browser without saving the changes.

Recover changes made in prior session
When enabled, this setting specifies that you want to use the same administrative configuration last used for the user's session. Recovers unsaved changes to the administrative configuration during the last user's session.

This field appears only if the user changed the administrative configuration and then logged out without saving the changes.

Administrative console areas and icons

Use the administrative console to create and manage resources, applications and servers or to view product messages.

To view the administrative console, ensure that the application server for the administrative console is running. Point a Web browser at the URL for the administrative console, enter your user ID and, if needed, a password on the Login page.

The console has the following main areas, which you can resize as desired:

Taskbar

The taskbar offers options for returning to the console Home page, saving changes to administrative configurations, specifying console-wide preferences, logging out of the console, and accessing product information.

Navigation tree

The navigation tree on the left side of the console offers links to console pages that you use to create and manage components in a WebSphere Voice Server administrative cell.

Click on a **+** beside a tree folder or item to expand the tree for the folder or item. Click on a **-** to collapse the tree for the folder or item. Click on an item in the tree view to toggle its state between expanded and collapsed.

WorkSpace

The workspace on the right side of the console contains pages that you use to create and manage configuration objects such as servers and resources. Click links in the navigation tree to view the different types of configured objects. Within the workspace, click on configured objects to view their configurations, run-time status, and options.

WebSphere Status

The status messages area at the bottom of the console provides information on messages returned by the WebSphere Voice Server. The information deals with problems in your administrative configuration as well as messages about run-time events. Click **Previous** or **Next** to toggle among displays of configuration problems and run-time events.

You can adjust the interval between automatic refreshes in the Preferences settings.

For details about administrative console buttons, page features, console navigation, and taskbar actions, refer to the WebSphere Application Server Network Deployment 5.1 information center using the links below. Note that not all the console features mentioned in the WebSphere Application Server information center are available in WebSphere Voice Server.

- Information on buttons
- Page features
- Console navigation
- Taskbar actions

For details specific to the WebSphere Voice Server administrative console, see Monitoring voice servers and resource status

Administrative console icons

The administrative console uses the following icons to indicate voice server status.

Icon	Meaning
⇨	Indicates a partial start of the server
⊗	Indicates a partial stop of the server
➡	Indicates that the server is running
✖	Indicates that the server is stopped
⊘	Indicates that the server status is unavailable
?	Indicates that the server status is unknown

Filtering on the administrative console

On the Voice Servers panel, there is a **Filter** option which allows you to filter the information by columns in the Overall Status table to best suit your needs.

Use these steps to filter on the administrative console:

1. Expand the Filter options by clicking the **+**.
2. Using the drop-down menu, select the column by which to filter. The options are **Voice Server Name** and **Status**.

   Note: The filter is case sensitive, so make sure that when you enter a filter value for Voice Server Name or Status, the cases you type match those used by the console. For example, the following Status values are accepted:

   - Started
   - Stopped
   - Partial Start
   - Unavailable
   - Unknown
3. Enter any filter criteria. The wildcards are: *,?,%.

Monitoring voice servers and resource status

The administrative console has specific panels for WebSphere Voice Server information.

Voice Servers panel

The Voice Servers panel is used to monitor the status of your voice servers. It is the default view when you log in to the administrative console.

There are two tables on the main Voice Servers panel, the Overall Status table and the WebSphere Status table.

Overall Status table

By viewing the Overall Status table, you can monitor the status of your entire voice server system, and you can filter the information in this table to suit your needs.

To start or stop a voice server, check the box beside the server you want to start or stop. Then click the **Start** or **Stop** button.

The following items are available for monitoring in the Overall Status table.

Overall Status
An icon depicts the overall status of your voice servers.

- If the automatic speech recognition (ASR) and text-to-speech (TTS) engines are started, the voice server is started.
- If the ASR and/or TTS engines are partially started, the voice server is partially started.
- If either the ASR or TTS engines are stopped and the other is not stopped, the voice server is partially started.
- If both the ASR and TTS engines are stopped, the voice server is stopped.

Total The number of voice servers in the current system.

Voice Server Name
This column displays the names of your voice servers using the convention `*node name*_server1`. Click a server link, such as dyn127-0-0-1_server1, to display detailed configuration and runtime status.

Status
An icon depicts the status of your individual voice servers. A Refresh icon updates the status icons for all voice servers when clicked.

WebSphere Status

Use the WebSphere Status area of the administrative console to view error and runtime event messages returned by WebSphere Voice Server. The WebSphere Status area displays along the bottom of the console and remains visible as you navigate from the Voice Servers panel to other pages.

The area displays two frames: **WebSphere Configuration Problems** and **WebSphere Runtime Messages**. Click **Previous** or **Next** to toggle between the frames.

Click the Refresh icon in the upper-right of the area to refresh the information displayed. You can adjust the interval between automatic refreshes in the Preferences settings.

WebSphere Configuration Problems
Displays the number of workspace files. This frame also displays the number of problems with the administrative configuration for the user ID. Click the number to view detailed information on the problems.

WebSphere Runtime Message
Displays the number of messages returned by WebSphere Voice Server as well as the number of error messages (**x** icon), warning messages (**!** icon), and informational messages (**i** icon). Click the number of messages to view details.

Configuration and runtime settings for specific voice servers

When you click a voice server link, such as `dyn127-0-0-1_server1`, in the Voice Servers panel, a new view displays detailed configuration and runtime status for that specific server. You can toggle between the **Configuration** and **Runtime** panels using the appropriately labeled tabs. Clicking any **Back** button will return you to the Voice Servers panel.

Configuration tab

This view lets you monitor and configure a specific voice server.

ASR Configuration
This table shows current configuration settings and controls for ASR resources.

For details about this table, refer to Configuring ASR settings.

TTS Configuration
This table shows current configuration settings and controls for TTS resources.

For more details, refer to Configuring TTS settings.

Runtime tab

This view shows the runtime status for a specific server. It is the visible or default interface when you click a specific voice server link.

General Status
This table provides general resource status for CPU and memory utilization, and it provides the overall status of the WebSphere Voice Server supporting components. The first column shows the

status item name, the second column provides a graphical display of the current setting, and the third column presents a short description of the status item

ASR Status

This table provides general resource status for ASR resources. The first column shows the status item name, the second column provides a graphical or text display of the current status, and the third column presents a short description of the status item. The three status items are **ASR Engines**, **Active Sessions**, and **Language installed**. The ASR engine status is shown using an icon. Click the ASR engine status icon to display the status name.

- If the ASR engine status icon is a filled-in green arrow (➡), the component is started.
- If the ASR engine status icon is a white arrow outlined in green (⇨), the component is partially started.
- If the ASR engine status is a red X (✖), the component is stopped.

If the number of active ASR sessions is less than the total, then the WebSphere Voice Server is not running to its full configuration capacity. If this number is always less than the total, then the total number of ASR engines could be reduced to save system resources. If this number is always equal to the total, check the logs to make sure a sufficient number of ASR engines are configured. You might need to increase the total number of configured ASR engines.

TTS Status

This table provides general resource status for TTS resources. The first column shows the status item name, the second column provides a graphical or text display of the current status, and the third column presents a short description of the status item. The four status items are **TTS Engines**, **Active Sessions**, **Default voice**, and **Voice(s) installed**. TTS engine status is shown using an icon. Click the TTS engine status icon to display the status name.

- If the TTS engine status icon is a filled-in green arrow (➡), the component is started.
- If the TTS engine status icon is a white arrow outlined in green (⇨), the component is partially started.
- If the TTS engine status is a red X (✖), the component is stopped.

If the number of active TTS sessions is less than the total, then the WebSphere Voice Server is not running to its full configuration capacity. If this number is always less than the total, then that the total number of TTS engines could be reduced to save system resources. If this number is always equal to the total, check the logs to make sure a sufficient number of TTS engines are configured. You might need to increase the total number of configured TTS engines.

Starting voice servers

Voice servers can be started using the administrative console or scripting.

Use the following steps to start a voice server in the administrative console:

1. In the Voice Servers panel, identify the voice server you would like to start (note the Status icon to determine the current status of the voice server) and click in the check box to the left of Voice Server Name.
2. Click the **Start** button. The status icon should appear as a green arrow pointing to the right (➡) when the voice server has started.

Alternatively, you can use the startVoiceServer wsadmin sample to start a single voice server from the command line using wsadmin.

Stopping voice servers

Voice servers can be stopped using the administrative console or scripting.

Use the following steps to stop a server in the administrative console:

1. In the Voice Servers panel, identify the voice server you would like to stop (note the Status icon to determine the current status of the voice server) and click in the check box to the left of Voice Server Name.
2. Click the **Stop** button. The status icon should appear as a red X (☒) when the voice server has stopped.

Alternatively, you can use the stopVoiceServer wsadmin sample to stop a single voice server from the command line using wsadmin.

Security

IBM WebSphere Voice Server uses the security infrastructure and mechanisms provided by WebSphere Application Server.

WebSphere Application Server allows administrators to protect a server using two security features: Global Security and Java 2 Security. These security features are managed through the Security panel on the administrative console.

Important: Java 2 Security cannot be enabled in order for WebSphere Voice Server to function properly. If you need security, enable Global Security in the WebSphere Voice Server administrative console. Do *not* select the option to enforce Java 2 Security.

For detailed information about the WebSphere approach to security, refer to Welcome to Security in the WebSphere Application Server information center.

Managing using scripting

WebSphere Voice Server provides sample wsadmin scripts for use with a single node installation.

Important: These scripts are intended to teach new WebSphere Voice Server administrators about potential uses of scripting and are not appropriate for use in production systems.

The WebSphere Application Server wsadmin tool provides the ability to execute scripts. You can use the wsadmin tool to manage a WebSphere Voice Server installation, configuration, application deployment, and server runtime operations. The WebSphere Application Server only supports Jacl, an alternate implementation of the interpreted script language Tcl (Tool Command Language), written in Java.

Note: Basic knowledge of the syntax for the Jacl scripting language is helpful in order to understand and modify the samples. For more information about Jacl, refer to Scripting: resources for learning.

These scripts are provided as samples to allow convenient access to common WebSphere Voice Server administrator tasks; specifically, starting and stopping voice servers, obtaining the voice server status, and backing up and restoring voice server configurations.

You must first issue the WebSphere Application Server **startServer** command, and then issue the **startVoiceServer** command.

Because each production environment is unique, these samples provide templates to assist new users in using wsadmin and Jacl. It is assumed that administrators will modify the code provided for their production environment (such as enabling WebSphere Application Server global security).

For more information on scripting, see Deploying and managing using scripting.

Sample scripts

The sample scripts assist with common administrator tasks that can be performed using wsadmin scripts. These samples are provided for convenience and to serve as examples. They include the following:

- **startVoiceServer**
- **stopVoiceServer**
- **statusVoiceServer**
- **backupVoiceServerConfig**
- **restoreVoiceServerConfig**

These sample scripts are installed into and are available from the `$WVS_ROOT/samples` directory.

startVoiceServer sample

On a base WebSphere Voice Server without Network Deployment, you must use the WebSphere Application Server **startServer** command line utility before running the startVoiceServer.jacl script. Use the **startServer** command as follows:

```
cd WAS_ROOT/bin
./startServer.sh server1
```

`WAS_ROOT` is where WebSphere Application Server is installed and *server1* is the name of your default WebSphere Application Server.

If WebSphere Voice Server has global security enabled, then sample utilities must be updated to use name and password parameters. This sample is in the `$WVS_ROOT/samples` directory.

To use this sample on Linux, issue the following:

```
./startVoiceServer.sh
```

The startVoiceServer script performs either one or both of the following actions:

- Starts the default WebSphere Application Server, server1.
- If the WebSphere Voice Server EAR is not started, this utility will start it.

Syntax

The sample's syntax is as follows:

```
startVoiceServer [option]
```

Parameters

The following option is available for the **startVoiceServer** sample:

--help
Prints a usage statement.

Related information

stopVoiceServer sample

statusVoiceServer sample

backupVoiceServerConfig sample

restoreVoiceServerConfig sample

stopVoiceServer sample

The **stopVoiceServer** sample stops the WebSphere Voice Server Enterprise Archive (EAR).

If WebSphere Application Server has global security enabled, then sample utilities must be updated to use name and password parameters. This sample is in the `$WVS_ROOT/samples` directory.

To use this sample on Linux, issue the following:

```
./stopVoiceServer.sh
```

This sample uses the stopVoiceServer.jacl script to perform the following actions:

- If the Voice Server EAR is started, this script will stop it.
- If the `-wvsonly` parameter is not specified, then the default WebSphere Application Server will be stopped.

For information on the WebSphere Application Server **stopServer** command, see the WebSphere Application Server Network Deployment 5.1 information center.

Syntax

The sample syntax is as follows:

```
stopVoiceServer [option]
```

Parameters

The following options are available for the **stopVoiceServer** sample:

-wvsonly
Stops only the WebSphere Voice Server EAR.

--help
Prints a usage statement.

Related information

startVoiceServer sample

statusVoiceServer sample

backupVoiceServerConfig sample

restoreVoiceServerConfig sample

statusVoiceServer sample

The **statusVoiceServer** sample obtains the current status of a voice server.

If WebSphere Application Server has global security enabled, then sample utilities must be updated to use name and password parameters. This sample is in the `$WVS_ROOT/samples` directory.

To use this sample on Linux, issue the following:

```
./statusVoiceServer.sh
```

This sample uses the statusVoiceServer.jacl script to perform the following actions:

- Queries the status of the Voice Server Enterprise Archive (EAR) managed bean (MBean), WVS51App. If the MBean exists, then the Voice Server EAR is started. If the MBean does not exist, then the Voice Server EAR is stopped.
- If started, then the WVSAdminControl MBean `getStatus` operation is queried for additional runtime status such as `Started, Stopped, Partial Start`.

Syntax

The sample syntax is as follows:

```
statusVoiceServer [option]
```

Parameters

The following option is available for the **statusVoiceServer** sample:

--help
Prints a usage statement.

Related information

startVoiceServer sample

stopVoiceServer sample

backupVoiceServerConfig sample

restoreVoiceServerConfig sample

backupVoiceServerConfig sample

The **backupVoiceServerConfig** sample script backs up the WebSphere Voice Server configuration and saves it to a zip file. This sample should only be run on a machine where the Voice Server has been installed.

If WebSphere Application Server has global security enabled, then sample utilities must be updated to use name and password parameters. This sample is in the *$WVS_ROOT*/samples directory.

This sample is located in the *$WVS_ROOT*/samples directory.

To use this sample on Linux, issue the following:

```
./backupVoiceServerConfig.sh
```

This sample uses the backupVoiceServerConfig.jacl script to perform the following actions:

- The Voice Server Enterprise Archive (EAR) will be stopped. This ensures that the configuration documents are not locked by the Voice Server and will not change while it is being backed up. You will have to restart the Voice Server EAR after the backup is complete.
- If the *backup_file* parameter (where *backup_file* is a name you have selected) is not specified, then the script creates a file in the current directory using the following name (overwriting the file if it exists already):

  WVSConfig_*yyyy*-*mm*-*dd*.zip where *yyyy* is the current year, *mm* is the current month and *dd* is the current day
- Uses the WVSAdminConfig managed bean (MBean) backup operation to store the Voice Server configuration into the **backup_file** (zip file format).

The **backupVoiceServerConfig** sample does not capture the product binaries or capture application data stored elsewhere (including on an HTTP Server). As such, the **backupVoiceServerConfig** utility is not a substitute for a comprehensive system backup facility.

Syntax

The sample syntax is as follows:

```
backupVoiceServerConfig [option]
```

Parameters

The following options are available for the **backupVoiceServerConfig** sample:

backup_file
An optional variable that specifies the location of the backup file to which information gets written.

--help
Prints a usage statement.

Related information

startVoiceServer sample

stopVoiceServer sample

statusVoiceServer sample

restoreVoiceServerConfig sample

restoreVoiceServerConfig sample

The **restoreVoiceServerConfig** sample restores the Voice Server configuration captured in a previous **backupVoiceServerConfig** for the same machine. You should only run this sample on a machine where the Voice Server has been installed.

If WebSphere Application Server has global security enabled, then sample utilities must be updated to use name and password parameters. This sample is in the `$WVS_ROOT/samples` directory.

This sample is in the `$WVS_ROOT/samples` directory.

To use this sample on Linux, issue the following:

```
./restoreVoiceServerConfig.sh
```

You must specify a backup zip file that was created from a captured in a previous **backupVoiceServerConfig** in the *backup_file* parameter for the same machine.

This sample uses the restoreVoiceServerConfig.jacl script to perform the following actions:

- The Voice Server Enterprise Archive (EAR) will be stopped. This ensures there are no sharing or read conflicts between the Voice Server and the restore processing. You will have to restart the Voice Server EAR after the restore is complete.
- Uses the WVSAdminConfig managed bean (MBean) restore operation to restore the Voice Server configuration from the *backup_file* (zip file format), overwriting the current configuration.

Syntax

The sample syntax is as follows:

```
restoreVoiceServerConfig [option]
```

Parameters

The following options are available for the **restoreVoiceServerConfig** sample:

backup_file
A required file name that specifies the location of the backup file to use for restoring the configuration.

--help
Prints a usage statement.

Related information

startVoiceServer sample
stopVoiceServer sample
statusVoiceServer sample
backupVoiceServerConfig sample

Automatically restarting server processes

There are several server processes related to WebSphere Voice Server products that the operating system can monitor and automatically restart when the server processes stop abnormally. This task describes how to set up monitored processes.

Before you begin

WebSphere Voice Server starts when WebSphere Application Server is started. If you set up WebSphere Application Server to start automatically, then WebSphere Voice Server will also start automatically. For more details, see Automatically restarting server processes in the WebSphere Application Server information center.

What you need to know

There are several environments where you might use this function of automatically restarting servers. Here is a list of processes you might consider restarting:

- The **server1** process on a standalone Application Server
- The **dmgr** process on a deployment manager node
- The **nodeagent** server process on any Application Server node in a deployment manager cell
- The **IBM HTTP Server** process
- The **IBM HTTP Administration** process

The installation wizard does not provide a way to create a service for a node agent because the deployment manager instantiates each node agent after installation when you add an Application Server node to the deployment manager cell. For this reason, you must manually create a function that automatically starts a failed nodeagent server process.

You must manually create a shell script that automatically starts any of the processes previously mentioned, on a Linux operating system.

In a Network Deployment environment, the **addNode** or **startNode** command starts a single unmonitored node agent only, the nodeagent process, and does not start all of the processes that you might define on the node. While running, the node agent monitors and restarts Application Server processes on that node, on a Linux platform. Each Application Server process has MonitoringPolicy configuration settings that the node agent uses when monitoring and restarting the process.

It is recommended that you set up a monitored node agent process manually through the rc.was example shell script on Linux platforms. The operating system monitors and automatically restarts the node agent process, nodeagent, if the process terminates abnormally, which means if the process stops without going through a normal shutdown. It is also recommended that you set up the deployment manager server, dmgr, as a monitored process. On a Linux platform, use the `rc.was` example shell script to set up the deployment manager dmgr server as a monitored process.

Steps for this task

After installing, set up a Linux shell script to automatically monitor and restart the nodeagent server or any other related server process:

1. Locate the `rc.was` example shell script in the *`WAS_ROOT`*`/bin` directory.

2. Create a new shell script for each process that the operating system is to monitor and restart.
3. Edit each shell script according to comments in its header, which provide instructions for identifying a WebSphere Application Server process.
4. Edit the inittab table of the operating system, to add an entry for each shell script you have created.

   Comments in the header of the `rc.was` file show a sample inittab entry line for adding the script. This inittab entry causes the Linux system to call each shell script whenever the system initializes. As it runs, each shell script monitors and starts the server process you specified.

Each shell script monitors and restarts an individual WebSphere Application Server process in a stand-alone environment, or a node agent or deployment manager process in an WebSphere Application Server Network Deployment environment.

Chapter 7. Tuning

Use the topic below to adjust the configuration and improve the performance of WebSphere Voice Server.

Tuning your configuration

Each installation will have a unique configuration. Tuning and capacity requirements are necessary to support each configuration.

For complete information on tuning WebSphere Application Server and IBM HTTP Server, see the Tuning section of the latest WebSphere Application Server Information Center.

The following parameter values were used successfully during product testing. Depending upon your installation configuration, these values might need updating to match your system configuration performance objectives. Use the following only as a guide for possible tuning values.

Tuning WebSphere Application Server

In most cases, the Java Virtual Machine (JVM) maximum heap should be 256 MB. This is the WebSphere Application Server default. If you are running a large number of automatic speech recognition (ASR) and text-to-speech (TTS) engines, then the JVM heap value should be 512 MB.

The JVM initial heap is 128 MB.

Tuning WebSphere Voice Server

Tuning the cwv.properties file to the following parameters is recommended:

- The RTSP Teardown parameter should be true (`com.ibm.voice.server.rtspbridge.oneteardown=true`)
- The number of ASR engines should equal the number of channels
- The number of TTS engines should equal one half the number of channels
- The RTSP Socket Timeout parameter should equal 150 (`com.ibm.voice.server.rtspbridge.sockettimeout=150`)
- The cache default expiration time should be set to 691200000 (`com.ibm.voice.server.rr.fetchexpires=691200000`). This large cache timeout prevents all the grammars with no set expiration from expiring at the same time.

Tuning Linux

Disabling the following Linux daemons is recommended:

- `service mdmpd stop`
- `service cups stop`
- `service crond stop`
- `service sendmail stop`
- `service rhnsd stop`

It is also recommended to set the following minimal values for Linux:

- `ulimit -n 4096`
- `net.core.rmem_default = 8388608`
- `net.core.wmem_default = 8388608`

- `net.core.rmem_max = 10485760`
- `net.core.wmem_max = 10485760`
- `net.ipv4.tcp_keepalive_time=30`

These values can be set using the native Linux sysctl application. For example:

```
sysctl -w net.core.rmem_max=10485760
```

Tuning SUSE 8.0

The pthread lib in SUSE 8.0 limits a process to only 1024 threads. To overcome this limitation, SUSE provides an updated pthread lib in the UnitedLinux Extension Pack for Lotus Domino.

For optimal tuning with SUSE and WebSphere Voice Server, complete the following steps:

1. Download and install the UnitedLinux Extension Pack for Lotus Domino..
2. Update the wvs.setenv script file provided by the WebSphere Voice Server installation to include the updated pthread lib. For example:

```
export LD_LIBRARY_PATH=/opt/dominowrap/domino6:$SPCH_ROOT/bin:$LD_LIBRARY_PATH
export LD_PRELOAD=/opt/dominowrap/domino6/libpthread.so:/opt/dominowrap/domino6/librt.so
```

Chapter 8. Troubleshooting

As an activity, troubleshooting or problem determination encompasses a wide range of tasks that might need to be performed at any phase in product usage. In addition to built-in preventative measures, the product provides a variety of tools to make problem determination easier.

The troubleshooting section of the documentation helps you understand why your voice server or product installation is not working, and helps you resolve the problem.

To effectively troubleshoot a voice server problem, use the following steps:

- Understand messages and alarms
- Understand log messages
- Enable and monitor trace
- Collect information for IBM Support

You may also troubleshoot according to component in your voice server and its environment, or according the particular task you are attempting (such as installation or administration).

Additional information on both approaches is also provided to streamline your troubleshooting process:

- Troubleshoot by component
- Troubleshoot by task type

Messages and alarms

The purpose of this section is to aid you in understanding WebSphere Voice Server messages.

Troubleshooting a deployed system involves collecting and analyzing different types of information from WebSphere Voice Server‘s many components. The types of information that are gathered and reported are as follows:

- **Informational messages** – Action is not required. Communication in this category is for advisory purposes.
- **Warning messages –** Recoverable error or event. You should review message, heightened application monitoring recommended, and (as necessary) take any action warranted to avoid future system malfunction.
- **Error messages –** Non-recoverable error or event. Action is required to correct problem. Full system function is non operational until the problem is corrected.
- **Alarms** – Messages that, besides being logged, are also reported as a console messages to system operator.

Several tools provided with WebSphere Voice Server provide messages and alarms to alert administrators about system events:

- **Administrative console** – Manage and view log and trace files (trace files must be configured to output in Basic or Advanced formats) and view alarms (console messages). The Voice Servers panel automatically refreshes every 60 seconds, but may be configured to refresh on a different schedule.
- **Log Analyzer tool –** View trace files (when trace files are in Log Analyzer format). The trace file name is `*WAS_ROOT*/logs/*server1*/trace.log`
- **WebSphere Application Server system log –** Errors, warnings, and informational messages. The log file is located at `*WAS_ROOT*/logs/*server1*/system.log`.

System logs cannot be disabled by a system administrator; they are always created and used anytime the system is operational. Trace logs can be enabled or disabled as needed by a system administrator or by technical support.

Log Analyzer

The Log Analyzer takes one or more service or activity logs, merges all of the data, and displays the entries.

Trace logs can be displayed in the Log Analyzer if the format of the trace log is the **Log Analyzer** format.

In the Log Analyzer, you can sort and group WebSphere Voice Server logs and traces by using the WebSphere Voice Server log and trace parameters.

For detailed information on the Log Analyzer tool, see the WebSphere Application Server Network Deployment 5.1 information center.

Monitoring log messages

WebSphere Voice Server can write system messages to several general purpose logs.

WebSphere Voice Server logs its messages to the WebSphere Application Server JVM log (`SystemOut.log`) and its trace messages to the WebSphere Application Server Trace (trace.log). Both files are located in *`WAS_ROOT`*`/logs/`*`server1`*`/`.

Comprehensive information about working with message logs may be found in the WebSphere Application Server Network Deployment 5.1 information center.

Related information

Message IDs
This section describes WebSphere Voice Servers message IDs, which use a 10-character format.

Identifier
This section describes the identifiers for WebSphere Voice Server.

Message IDs

This section describes WebSphere Voice Servers message IDs, which use a 10-character format.

CWV*SSNNNNT*

Where:

CWV = IBM-registered three-character prefix identifying WebSphere Voice Server. The elements of the CWV code appear below:

- *SS* = Used to identify the WebSphere Voice Server components within the product. See the chart below for more information.
- *NNNN* = Numeric message ID (4-digits).
- *T* = Severity type (**I**nformation, **W**arning, or **E**rror).

Click a component identifier below to see all the messages for that component.

Note: The *message number* is any 4-digit number associated with the log message.

Table 13. Component log messages

Component identifier	**Component**
CWVAE	ASR Engine

Table 13. Component log messages (continued)

Component identifier	Component
CWVAB	ASR Bean
CWVAC	ASR Control Interface
CWVRB	Recognizer
CWVTE	TTS Engine
CWVSB	Synthesizer
CWVSM	Session Manager
CWVMD	MRCP Message Dispatcher
CWVGM	Grammar Manager
CWVRT	RTSP Bridge
CWVMC	Media Converter
CWVRR0000 - 4999 CWVRR5000 - 9999	Resource Resolver
CWVGC	Grammar Compiler
CWVPT	Product Tools
CWVSY	Systems Management
CWVCC	Common

Related information

Identifier
This section describes the identifiers for WebSphere Voice Server.

Identifier

This section describes the identifiers for WebSphere Voice Server.

WebSphere Voice Server and the MRCP client (such as the VoiceXML Browser) create unique identifiers in order to exchange information. You can see examples of these identifiers in the following sample `SystemOut.log` message:

```
[6/16/04 19:31:53:430 EDT] 266a8ddf Synth E
CWVSB0004E: [sid=4442.IBM.127.0.0.1][rid=13][ltag=18853:18853]
MRCP SPEAK request has failed. Reason: No resource available: 4442.IBM.127.0.0.1
```

[sid] Stands for MRCP **session-id**. WebSphere Voice Server creates this unique identifier for the session and passes it to the MRCP client.

In the example above, the [sid] is `4442.IBM.127.00.0.0.1`.

[rid] Stands for MRCP **request-id**. The MRCP client creates this unique identifier for the request and passes it to WebSphere Voice Server.

In the example above, the [rid] is 13.

[ltag] Stands for MRCP **logging-tag**. The MRCP client creates this unique identifier and passes it to WebSphere Voice Server.

In the example above, the [ltag] is `18853:18853`.

[turn] Stands for the WebSphere Voice Server turn identifier. WebSphere Voice Server creates and manages this identifier. The WebSphere Voice Server MRCP Synthesizer and Recognizer components count the number of “Recognize” or “Speak” MRCP messages. This identifier starts at 1.

In the example above, no information was available for [turn], thus the log does not specify that identifier.

These identifiers allow for tracking the MRCP activity throughout the system on a per-session and per-request basis. If any of the four fields are not specified, then the information was not available. These identifiers are readily available to match up with the logs created by the MRCP client.

Related information

Message IDs
This section describes WebSphere Voice Servers message IDs, which use a 10-character format.

Enabling and monitoring trace

Use trace to obtain detailed information about the execution of WebSphere Voice Server components.

WebSphere Voice Server logs its traces to a trace log located at `WAS_ROOT/logs/server1/`. The trace log shows trace events such as function entry/exits, component events, and debug. Two topics provide additional information to guide system administrators:

- Enabling trace
- Monitoring trace

Enabling trace

Trace for a voice server process is enabled by using the administrative console. You can configure the voice server to start in a trace-enabled state by setting the appropriate configuration properties.

In the administrative console, enable and configure trace by using the following steps:

1. Expand Troubleshooting in the navigation panel by clicking the **+**.
2. Click **Logs and Trace**.
3. Click the name of the server, such as server1, listed in the Server column of the table.
4. Click **Diagnostic Trace**.

When you click the **Modify** button under the Trace Specification section of the Configuration tab, only three of the trace groups displayed in the Modify Trace panel are relevant to WebSphere Voice Server. They are the following.

IBM WVS
Includes all of the basic WebSphere Voice Server components.

IBM WVS ASR
Includes all of the ASR components.

Table 14. ASR components

Acronym	Description
ASRAPI	SMAPI Messages
ASRAUD	Audio Library
ASRAW	Add Word
ASRBEAN	ASR Bean
ASRCONF	Confidence Scoring
ASRCTRL	Control Interface
ASRDM	Detailed Match
ASRENG	ASR Engine (configuration and recognition results)
ASRFM	Fast Match

Table 14. ASR components (continued)

Acronym	Description
ASRLAT	Latency
ASRLM	Language Model
ASRNBST	N-Best results
ASRSEQ	Program Execution Sequence
ASRSIG	Signal Processing
ASRSRCH	Stack Search
ASRTRN	Training
ASRVOC	Vocabulary and Pronunciations
ASRWSI	Work Station Interface
RECOGNIZER	Recognizer Bean

IBM WVS TTS

Includes all of the TTS components.

Table 15. TTS components

Acronym	Description
CMM	Concatenative Memory Manager
DIC	Dictionary
ECI	ECI Interface API
ETI	Common Development Library
MCL	Media and Control Layer
MLD	Multilevel Data Structure
MOR	Morphological Analyzer
NOR	Normalizer
PAR	Language Parser
PHO	Phonetizer
PRE	Preprocessor
PRO	Prosody
SEM	Semaphores
SOU	Sound
SSM	XML Parser, SSML Processor and Filter
SYL	Syllable Analyzer
TOR	Torrent

Important: Enable IBM WVS ASR or IBM WVS TTS tracing only if advised by IBM Support. Enabling these trace groups will cause performance issues.

Because WebSphere Voice Server uses the common WebSphere approach to trace, the WebSphere Application Server Network Deployment 5.1 information center contains detailed information on configuring trace in the administrative console.

Trace Output Formats, the last configuration parameter shown on the console, allows you to specify one of three levels for trace output:

Basic (Compatible)
Preserves only basic trace information. Select this option to minimize the amount of space taken up by the trace output. The results are sent to a text file.

Advanced
Preserves more specific trace information. Select this option to see detailed trace information for use in troubleshooting and problem determination. The results are sent to a text file.

Log Analyzer
Preserves trace information in a format that is compatible with the Log Analyzer tool. Select this option if you want to use the trace output as input to the Log Analyzer tool. The results are sent to a binary file.

Monitoring trace

WebSphere Voice Server uses the following log parameters:

- Product = WVS
- Manufacturer = IBM
- LongName(ComponentId) = IBM WVS [component name]
- ShortName(SourceId) = [component name]

To display the trace log in the Log Analyzer, select the **Log Analyzer** format from the Trace Log Output pull-down menu on WebSphere Application Server administrator console.

Isolating audio problems between WebSphere Voice Server and the third-party gateway

Sometimes it is difficult to isolate audio problems between WebSphere Voice Server and the third-party gateway. The first step toward problem determination is to ensure that the correct TTS audio is produced by the Voice Server. To do this, you can listen to the generated .pcm file for each TTS synthesis request. If the quality of the TTS audio is degraded from what you hear in the .pcm file, then the problem is most likely with the gateway.

You can also listen to the .pcm files (using a tool of your choosing) for each ASR request to ensure that the correct ASR audio is produced. IBM WVS MediaCov records the audio as it is received from the gateway. The ASR engine records pre- and post-endpointed audio. This allows you to hear if the ASR audio is degraded by the gateway.

If **debug** is enabled for certain components, these components will log trace messages that reference binary or text files. Certain trace messages reference the name of the binary or text file. For example, if **debug** event is enabled for the IBM WVS MediaConv managed bean (MBean), then the Media Converter will create audio files to record all of the audio passed.

These additional files, including the .pcm files, are located in a WebSphere Voice Server-specific directory: `WAS_ROOT/logs/server/CWV/time-stamp/`

Each time that WebSphere Voice Server is restarted, a new *time-stamp* directory is created so that no existing files are overwritten.

Enabling **debug** for these components creates a lot of files on disk. You should only perform this activity at the direction of IBM Support or to investigate a specific audio problem. After you complete this activity, run the Collector Tool. The Collector Tool will also delete all information out of the **CWV** directory created when **debug** is enabled.

Table 16. Additional log files

Component	Subcomponent	Files Created	File Naming Convention
IBM WVS ASR - ASR engine	If the trace level for subcomponent ASRAUD is set to entry/exit, then post-endpointed audio is saved.	All utterances for all sessions are recorded. These recordings are endpointed audio.	`<SessionID>.<TurnID>.ep.pcm`
IBM WVS ASR - ASR engine	If the trace level for subcomponent ASRAPI is set to entry/exit, then pre-endpointed audio is saved.	All utterances for all sessions are recorded. These recordings are not endpointed audio (that is, they contain extra silence).	`<SessionID>.<TurnID>.pcm`
IBM WVS ASR - ASR engine	ASRTRN	Saves data for site adaptation.	`<SessionID>.<TurnID>.ep.`*`xxx`* (where *xxx* can be `bsf`, `cep`, or `tag`.

IBM WVS Media Converter	N/A	All audio for all sessions is recorded, meaning, the audio before it is passed to the ASR engine and the audio after it is received from the TTS engine.	`<SessionID>.mc.asr.pcm`
			or `<SessionID>.<TurnID>.mc.asr.pcm` if the save-waveform header is enabled in the MRCP RECOGNIZE request
			`<SessionID>.<TurnID>.mc.tts.pcm`

Related information

Identifier
This section describes the identifiers for WebSphere Voice Server.

Working with trace

Note: The sections “Enabling trace on an application client or stand-alone process” and “Enabling trace on client and stand-alone applications” do not apply to WebSphere Voice Server, version 5.1.

Collecting information for IBM Support

The collector tool gathers extensive information about your WebSphere Voice Server and WebSphere Application Server installation and packages it in a Java archive (JAR) file that you can send to IBM Customer Support to assist in problem determination. Information in the JAR file includes logs, property files, configuration files, operating system and Java data, and the absence or level of each software prerequisite.

The collector tool runs to completion despite any errors that it might find, such as missing files or commands. The collector tool collects as much data in the JAR file as possible.

There are several steps to follow in order to run the collector tool.

Run the collector command to create the JAR file if IBM Support needs more information to solve your problem.

To run the collector summary option, start from a temporary directory outside of the WebSphere Application Server product installation root directory and enter the following commands.

```
>mkdir work
>cd work
>/opt/WebSphere/AppServer/bin/collector.sh - Summary
```

To create a lightweight version of the information in a text file and on the console, see the WebSphere Application Server Network Deployment 5.1 information center. The lightweight information is useful for getting started in communicating your problem to IBM Support.

Additional information

Detailed information about running the Collector tool may be found in the WebSphere Application Server Network Deployment 5.1 information center.

Running the collector tool

Use the **collector** command to create the JAR file.

Steps for this task

1. Log on to the system as **root**.

   The Collector Tool should be run under the root or administrator user ID because some of the commands to be executed require system access authority.
2. Verify that Java 1.4.1 or higher is available in the path.

   The collector program requires Java code to run. Verify that all necessary information is in the path being used by the collector program and that you are not running the program from within the WebSphere Application Server product installation root directory. Verify that the path contains the following:
   - `/bin`
   - `/sbin`
   - `/usr/bin`
   - `/usr/sbin`
3. Create a work directory where you can start the collector program.
4. Make the work directory the current directory.

   The collector program writes its output JAR file to the current directory. The program also creates and deletes a number of temporary files in the current directory. Creating a work directory to run the collector program avoids naming collisions and makes cleanup easier. You cannot run the collector tool in a directory under the WebSphere Application Server installation directory.
5. Run the collector program by entering the **collector** command from the command line.

   Using the **collector** command with no additional parameters gathers one copy of the node data and data from each server in the node, and stores them in a single JAR output file.

Note: Set the path correctly to use the non-qualified version of the command. Please note, `WAS_ROOT/bin` must be in the path to locate the **`collector.sh`** command.

The WebSphere Application Server installation root directory is determined at installation. It is identified in the `setupCmdLine.sh` file.

You can enter a fully qualified path to the collector command. For example, enter these commands in a default installation:

```
>mkdir work
>cd work
>/opt/WebSphere/AppServer/bin/collector.sh
```

Results

The name of the JAR file is based on the hostname and package of the Application Server product, in the format: `hostname-ND-WASenv.jar` or `hostname-Base-WASenv.jar`. For example, if you run the collector tool on the ws-laceweb server within a Network Deployment cell, the filename is `ws-laceweb-ND-WASenv.jar` .

The Collector.log file is one of the files collected in the `hostname-ND-WASenv.jar` or `hostname-Base-WASenv.jar` file.

What to do next

Send the `hostname-ND-WASenv.jar` or `hostname-Base-WASenv.jar` file to IBM Support for analysis.

Important: Except for any information documented in writing as confidential between the parties, all information provided to IBM is non-confidential. Please review all collected data and remove or discard any information not to be provided to IBM in accordance with the foregoing. Any information or other materials provided to IBM through this utility is done at your own risk.

Troubleshooting by component: What is not working?

This section provides troubleshooting information based on the task you were trying to accomplish when the problem occurred.

To find more information about your problem, select a task category from the list below.

If you do not see a task that resembles yours, or if the information provided does not solve your problem, contact IBM Support for further assistance.

Installation troubleshooting tips

If you are having problems installing the WebSphere Voice Server, follow these steps to resolve the problem.

- If possible, follow the steps outlined in Troubleshooting the installation.
- Browse the relevant log files for clues:
  - The main installation log file: `install_dir/log/CWVinstall.log`
  - IBM Http Server log: `WAS_ROOT/logs/ihs_log.txt`
  - IBM TTS log: `/var/opt/IBM/ibmtts/log/install.log`
  - WebSphere Application Server logs: `WAS_ROOT/logs/server1`
- Ensure that you have installed the correct level of dependent software, such as the operating system version and revision level, by reviewing the Identifying hardware and software requirements section of the information center.

If none of these steps solves the problem, check to see if the problem is identified and documented using the links in Diagnosing and fixing problems: Resources for learning. If you do not see a problem that resembles yours, or if the information provided does not solve your problem, contact IBM support for further assistance.

For current information available from IBM Support on known problems and their resolution, see the IBM Support page. IBM Support has documents that can save you time gathering information needed to resolve this problem. Before reporting this problem, see the IBM Support page.

Administration and administrative console troubleshooting tips

This topic contains troubleshooting tips for administration and the administrative console.

For detailed information, see the WebSphere Application Server Network Deployment 5.1 information center.

Security troubleshooting tips

This topic contains troubleshooting tips for security components.

Important: Java 2 Security must be disabled when using WebSphere Voice Server to avoid performance degradation.

For detailed information, see the WebSphere Application Server Network Deployment 5.1 information center.

TTS troubleshooting tips

This topic contains troubleshooting tips for text-to-speech (TTS).

- The TTS engine(s) will not start
- Speech Synthesis Markup Language (SSML) or lexicon-related troubleshooting tips
- The embedded audio or lexicon document cannot be fetched, or the embedded audio file within the SSML document will not play
- The SSML document encounters a parse error
- When the <voice> tag has a gender attribute set to ″neutral″, the voice is not changed. All sentences are read with the same voice and pitch as previously specified
- A wave file is truncated

The TTS engine(s) will not start

Use the following checklist to troubleshoot this problem:

- Run the following command: **rpm -qa | grep ibmtts**

  The language you are trying to use should be installed.
- Make sure you have installed the correct language. To do this, navigate to `/var/opt/IBM/ibmtts/cfg/eci.ini` and look for the following entries: [1.0], [1.1], [8.0], [6.0].

  The [1.0] is for American English, the [1.1] is for British English, the [8.0] is for Japanese, and the [6.0] is for Chinese.
- Check the *WAS_ROOT*/logs/*server1*/SystemOut.log for errors.

SSML or lexicon-related troubleshooting tips

- Is your SSML compliant with SSML 1.0 standards?
- Have you validated your lexicon file using the Voice Toolkit Lexicon editor? For more information on the Lexicon editor provided in Voice Toolkit for WebSphere Studio, see the Voice Toolkit information center on the Voice Toolkit CD. In order to access the Voice Toolkit information center, you must have the Voice Toolkit installed.
- Check the *WAS_ROOT*/logs/*server1*/SystemOut.log to see if any errors have been logged.

The embedded audio or lexicon document cannot be fetched, or the embedded audio file within the SSML document will not play

- Make certain that the HTTP server is up and running on the host system:
  1. Change directory to `/opt/IBMHttpServer/bin` and type **./apachectl start**.
  2. Then check to make sure that the server is started by running the following command: **ps auxw | grep httpd**.

     If the server is running, you should see something similar to the following:

```
root     19962  5.4  0.2 10320 6964 ?       S    13:44   0:00
opt/IBMHttpServer/bin/httpd -d /opt/IBMHttpServer
nobody   19963  0.0  0.2 10320 6988 ?       S    13:44   0:00
opt/IBMHttpServer/bin/httpd -d /opt/IBMHttpServer
nobody   19964  0.0  0.2 10320 6988 ?       S    13:44   0:00
opt/IBMHttpServer/bin/httpd -d /opt/IBMHttpServer
nobody   19965  0.3  0.2 10320 6988 ?       S    13:44   0:00
opt/IBMHttpServer/bin/httpd -d /opt/IBMHttpServer
nobody   19966  0.0  0.2 10320 6988 ?       S    13:44   0:00
opt/IBMHttpServer/bin/httpd -d /opt/IBMHttpServer
nobody   19967  0.0  0.2 10320 6988 ?       S    13:44   0:00
opt/IBMHttpServer/bin/httpd -d /opt/IBMHttpServer
root     19969  0.0  0.0  3696  648 pts/1   S    13:44   0:00
rep httpd
```

- Is the path you specified correct for the file you want?
- Check the *WAS_ROOT*/logs/*server1*/SystemOut.log to see if any errors have been logged.
- Does the file exist in the location specified?
- Verify that the file is in the correct audio format. VoiceXML specifications require that a platform support the following playing and recording audio formats:
  - Raw (headerless) 8kHz 8-bit mono mu-law [PCM] single channel.
  - Raw (headerless) 8kHz 8 bit mono A-law [PCM] single channel.
  - WAV (RIFF header) 8kHz 8-bit mono mu-law [PCM] single channel.
  - WAV (RIFF header) 8kHz 8-bit mono A-law [PCM] single channel.

Also, if the SSML processor cannot find the audio file, it will synthesize the files that it can find. This means that there is no error message rendered, but you will not hear the file played. In order to troubleshoot this error you will need to turn on tracing to see why the file was not fetched.

The SSML document encounters a parse error

- Is your SSML compliant with SSML 1.0 standards?
- Have you validated your lexicon file using the Voice Toolkit Lexicon editor? For more information on the Lexicon editor provided in Voice Toolkit for WebSphere Studio, see the Voice Toolkit information center on the Voice Toolkit CD. In order to access the Voice Toolkit information center, you must have the Voice Toolkit installed.
- Check the *WAS_ROOT*/logs/*server1*/SystemOut.log to see if any errors have been logged.

When the <voice> tag has a gender attribute set to "neutral", the voice is not changed

Certain <voice> tag attributes, such as gender ″neutral″ and age, are not supported for any concatentive voices, including US English and Chinese voices. Also note that the ″variant″ attribute is not supported for the US English and Chinese voices.

A wave file is truncated

The amount of text included in one speak request is limited to the amount of text that can be spoken in five minutes. The synthesized audio for larger amounts of text may be truncated.

ASR troubleshooting tips

This topic contains troubleshooting tips for automatic speech recognition (ASR).

General

- The ASR engine(s) will not start

Note: Audio and compiled grammars received by the ASR engine can be saved on the server by setting the IBM WVS ASR - ASRAPI trace level to medium (entry/exit). See the additional log files table for more information.

Grammar or lexicon

- A grammar has compilation errors
- A grammar or lexicon cannot be fetched
- Compilation of a large grammar or a large lexicon is slow and times out

Recognition

- The environment is noisy
- Large grammar issues
- No-input timeout is happening too often
- No-match is happening too often
- How do I increase the accuracy of my grammar?
- False barge-ins are occurring

The ASR engine(s) will not start

This problem could be caused by a couple of issues:

- The ASR engines will not start if installation verification test (IVT) fails. Check the `$WVS_ROOT/logs/server_name/SystemErr.log` and `SystemOut.log` for clues. Also refer to the information center section on troubleshooting IVT.
- The ASR engines will not start if they cannot load their dynamic libraries. Linux will not be able to load the libraries if the LD_LIBRARY_PATH is not properly set. This path is set when you run the wvs.setenv script.

If this information does not solve your problem, see the IBM Support page for further assistance.

A grammar has compilation errors

Use the Voice Toolkit for WebSphere Studio or the command-line voiceTest tool to extract the compilation messages from the package file. If the messages indicate that the problem is caused by missing baseforms, then use the Voice Toolkit to generate the appropriate baseforms using the lexicon editor.

A grammar or lexicon cannot be fetched

Use a browser to check make sure that the grammar or lexicon can be viewed successfully. Also be sure to check for typos in the grammar or lexicon uniform resource identifier (URI).

Compilation of a large grammar or a large lexicon is slow and times out

Use the Voice Toolkit for WebSphere Studio or the command-line voiceTest tool to see if the file has a lot of auto-added pronunciations. If so, add to these pronunciations to the lexicon. Preload the lexicon when the server starts using the configuration file.

The environment is noisy

Adjust the sensitivity level. Lower the level if the background noise/speech is being recognized or causing barge-ins.

Large grammar concerns

Use the Voice Toolkit for WebSphere Studio or the command-line voiceTest tool to generate a lexicon and see if any words are being auto-added. Make sure that the spellings and pronunciations are correct.

Also see if the grammar accepts too many phrases (use **fsgenum** to list acceptable phrases). There is a limit of one million pronunciations per grammar, so you may need to keep to fewer than 500,000 different words.

If a grammar is not compiling, and you have tracing turned on, you may see an out of memory error similar to the following in the trace.log:

```
[6/22/04 13:49:56:702 EDT] 257e78a2 ExceptionUtil E CNTR0020E:
Non-application exception occurred while processing method "defineGrammar" on
bean "BeanId(WVS5.1-kirkwood#wvsejb.jar#RecognizerEJB, null)".
Exception data: java.lang.OutOfMemoryError
[6/22/04 13:49:56:820 EDT] 257e78a2 LocalTranCoor E WLTC0017E:
Resources rolled back due to setRollbackOnly() being called.
[6/22/04 13:49:56:881 EDT] 257e78a2 Dispatcher    E   CWVMD0003E:
[sid=0.IBM.9.2.184.225][rid=100][turn=1] Failed to process recognizer request.
Reason:  ; nested exception is: com.ibm.ws.exception.WsEJBException:
nested exception is: java.lang.OutOfMemoryError
```

Or, if you are trying to compile a grammar using voiceTest.sh and it takes 10 minutes or more to compile, you may see an error similar to the following:

```
[root@kirkwood /opt/WebSphere/VoiceServer/lib]$ ./voiceTest.sh
define -grammar /opt/WebSphere/VoiceServer/lib/busLssi.bnf
IBM WebSphere Voice Server voiceTest

[04/06/22 13:37:06:928 EDT] main     pt           A CWVPT0027I: Connecting to localhost 554...
[04/06/22 13:37:07:226 EDT] Thread-1 pt           A CWVPT0029I: Waiting for reply...
[04/06/22 13:37:07:227 EDT] Thread-1 pt           A CWVPT0030I: RTSP session: "0.IBM.127.0.0.1"
[04/06/22 13:37:07:230 EDT] main     pt           A CWVPT0028I: RTP contact at 127.0.0.1 50002
[04/06/22 13:49:56:904 EDT] main     pt           E CWVPT0010E: RTSP error 500 Internal Server Error
```

These errors indicate that you need to increase the WebSphere Application Server Maximum Heap Size setting. The default setting is 256 MB.

To increase the Maximum Heap Size setting, do the following:

1. Launch and log in to the administrative console.
2. In the left navigation pane, select **Servers**, and then select **Application Servers**.
3. Click the name of the server you are using. By default, it is server1.
4. On the server panel, in the Additional Properties table, click **Process Definition**.
5. On the Process Definition panel, in the Additional Properties table, click **Java Virtual Machine**.
6. On the Java Virtual Machine panel, in the Configuration table, increase the value of the Maximum Heap Size in MB.
7. Click the **Apply** button at the bottom of the Configuration table to save your changes to the master configuration.
8. Restart WebSphere Application Server for the changes to take effect.

No-input timeout is happening too often

Check audio level. Raise the audio gain if it is low, or raise the sensitivity level.

No-match is happening too often

Use the following checklist to troubleshoot this problem:

- Make sure the audio matches the pronunciations of the words in the grammar.
- Make sure that the timeouts are not too short. If so, then a phrase may be truncated by a short pause.
- Make sure that audio gain is not too high. If so, then the signal is clipped.
- Make sure that audio is not distorted by the voice activity detection in the gateway (audio with perfect artificial silence).
- Adjust the speed-accuracy knob towards the ″accurate″ level (but note that this will impact speed performance).

How do I increase the accuracy of my grammar?

Use the following checklist to troubleshoot this problem:

- Check the spelling of your words and minimize use of abbreviations when possible. For example, use ″Street″ instead of ″St.″ and ″Road″ instead of ″Rd.″
- Break long compound words or numbers into separate words if are difficult to say without pausing.
- Check the coverage of the grammar. For example, listen to recordings of live interactions and make sure the grammar covers what is being said.
- Remove nonsensical or low-probability phrases that can cause confusion. Enumerate all phrases the grammar will accept and redesign if there are too many.
- Adjust the weights of rules. See the Speech Recognition Grammar Specification (SRGS) 1.0 at http://www.w3.org/TR/speech-grammar/.

False barge-ins are occurring

Use the following checklist to troubleshoot this problem:

- Make sure the echo cancellation is activated correctly.
- Lower the sensitivity level.
- Reduce the volume of the prompt.

Third-party gateway communication problems

This topic contains troubleshooting tips for communication problems for third-party gateways.

Ensure that the following RTSP parameters are set correctly in the gateway:

- Port configuration parameter
- Synthesizer media parameter
- Recognizer media parameter

Troubleshooting by task: What are you trying to do?

This section provides troubleshooting information based on the task you were trying to accomplish when the problem occurred.

To find more information about your problem, select a task category from the following list.

If you do not see a task that resembles yours, or if the information provided does not solve your problem, contact IBM Support for further assistance.

Troubleshooting installation problems

Select the problem you are having with the WebSphere Voice Server installation:

- Installation either completes with errors or warnings, or hangs.
- The installation process completes, but the application server or deployment manager does not start or starts with errors.
- The installation completes, but the WebSphere Voice Server installation verification does not work.
- There are error messages issued during installation.
- The installation completes but the administrative console does not start.

If you do not see a problem that resembles yours, or if the information provided does not solve your problem, see Troubleshooting the Installation.

For current information available from IBM Support on known problems and their resolution, see the IBM Support page. IBM Support has documents that can save you time gathering information needed to resolve this problem. Before opening a PMR, see the IBM Support page.

Errors while installing the product

If the WebSphere Voice Server installation program indicates that errors were encountered while installing the product, do the following:

1. Browse all of the log files in `$WVS_ROOT/logs/` for clues. Pay particular attention to the main installation log file, `CWVinstall.log`.
2. Check the command prompt from which the installation panel was launched for error messages.
3. Look up any error or warning messages in the message reference table.
4. Review Troubleshooting the Installation.

Installation completes but IVT does not work

If the WebSphere Voice Server installation program completes successfully, but the Installation Verification Test (IVT) does not run, do the following:

1. Browse the application server log files `$WVS_ROOT/logs/server_name/SystemErr.log` and `SystemOut.log` for clues.
2. View the JVM logs of the hosting application server for clues, after attempting to run WebSphere Voice Server IVT.
3. Look up any error or warning messages in the message reference table.
4. Verify any security-related problems.
5. Verify that `$SPCH_ROOT` is set. If it is not, then it is likely that `/wvs.setenv` was not executed.
6. Review Troubleshooting the Installation for more information.

If you do not see a problem that resembles yours, or if the information provided does not solve your problem, check to see if your problem has been identified. See Diagnosing and fixing problems: Resources for learning for information on getting the latest updates. If your problem has not been reported, see the IBM Support page for further assistance.

Troubleshooting testing and first time run problems

Select the problem you are having with testing WebSphere Voice Server:

- The application server or deployment manager does not start or starts with errors.
- The application does not start or starts with errors.
- Access problems after enabling security.
- Errors after enabling security.
- Errors after enabling Secure Sockets Layer.
- Errors in messaging.

If you do not see a problem that resembles yours, or if the information provided does not solve your problem, see IBM Support page.

The IBM Support site provides current information available on known problems and their resolutions. IBM Support has documents that can save you time gathering information needed to resolve this problem. Before opening a Problem Management Report (PMR), see the IBM Support page.

Troubleshooting application runtime and management problems

This topic contains troubleshooting tips for application runtime and management problems.

For detailed information, see the WebSphere Application Server Network Deployment 5.1 information center.

Chapter 9. Resources for learning

Use the following links to learn more about the technologies related to IBM WebSphere Voice Server for Multiplatforms Version 5.1:

- WebSphere Application Server products
- RedHat Linux Web site
- SUSE LINUX Web site
- Sun Microsystems Web site
- Java specification documents