Multimodal Tools V4.1.2 / 4.1.2.2
Frequently Asked Questions

Including:
Multimodal Toolkit V4.3.1 / 4.3.2 for WebSphere® Studio
Multimodal Browser V4.1.2

July 2004

1. What does "multimodal" mean?

The content of a multimodal application specifies the presentations and interactions in both visual and voice modalities. In most cases, users can choose the most efficient input method when using their mobile devices, which might have limited input, output, and display capabilities.

The Multimodal Toolkit lets you use standards-compliant combinations of existing languages to create a multimodal application by adding snippets of speech markup to the XHTML™ visual markup. The voice portion of a multimodal application uses VoiceXML, providing an easy way to manage a voice dialog between a user and a speech recognition engine. XHTML+Voice, informally known as "X+V," is the markup language used to create multimodal applications.

Mobile devices are targeted for particular uses. They support the features they need for the functions they are designed to fulfill. A multimodal authoring system lets you speech-enable the visible elements of a visual interface, which has particular importance for mobile devices.

2. What is XHTML+Voice?

XHTML+Voice, or X+V for short, is a markup language for multimodal Web pages. With X+V, Web developers can create Web pages that let end-users use voice input and output as well as traditional visual (GUI) interaction. X+V does this by providing a simple way to add voice markup to XHTML. Hence the name "XHTML plus Voice."

X+V fits into the Web environment by taking a normal visual Web user-interface, and speech-enabling each part of it. That is, if you take a visual interface and break it up into its basic parts (such as an input field for a time of day, a check box for AM or PM, and so on), you can then simply enable the use of voice by adding voice markup to the visual markup. X+V consists of visual markup, a collection of snippets of voice markup for each element in the user interface, and a specification of which snippets to activate when. For visual markup, X+V uses the familiar XHTML standard. For voice markup, it uses a (simplified) subset of VoiceXML. For associating the snippets of VoiceXML and user-interface elements, X+V uses the XML Events standard. All of these are official standards for the Web as defined by the Internet Engineering Task Force (IETF) that governs web standards.

Refer to the **XHTML+Voice 1.2 specification** at the following Web site:
http://www.voicexml.org/specs/multimodal/x+v/12/spec.html

For more information, refer to the following Web sites and specifications:

- World Wide Web Consortium® (W3C) Web site:
  http://www.w3.org/
- Online tutorials in these skills:
  http://www.w3schools.com/
- XML Events specification:
  http://www.w3.org/TR/xml-events/

3. How does this relate to the W3C multimodal activity/working group?

Motorola®, Opera Software ASA, and IBM submitted the XHTML+Voice specification to the W3C multimodal working group in January of 2002. IBM participates in the working group along with other interested industry participants. X+V is under consideration by the working group, whose mission is to create specifications for the multimodal Web.

4. Is XHTML+Voice an open standard?

Motorola, Opera Software ASA, and IBM submitted to the W3C a proposal for an XHTML+Voice open standard. XHTML+Voice is based on XHTML, VoiceXML, and XML Events. All of these are open standards.

5. What are the advantages of XHTML+Voice?

Web applications built on XHTML+Voice can be accessed by voice devices, browser-based devices, and new multimodal devices. This means:

- Multimodal interaction can help improve the usability of data services, such as corporate applications, contact centers, financial accounts, traffic and weather reporting, restaurant and movie guide, sports, news room, messaging, horoscope, lottery, and many more. You can interact with the application using a keyboard/stylus while in a meeting or by voice when appropriate.
- Multimodal applications can give users the flexibility to choose the mode of interaction that is the most suitable for the task at the given moment. This may include the use of multiple modes of communication to give an enhanced user experience. For example, the user can alternate between speaking and tapping in the interface.
- As devices continue to get smaller, multimodal interaction can help increase the effectiveness of the device by combining multiple input and output modes of communication.
- The total cost of ownership can be lower than having a telephony infrastructure and an Internet infrastructure. Also the development and maintenance cost may be reduced.

6. What products support XHTML+Voice?

IBM is extending its WebSphere line of products to support XHTML+Voice. The Multimodal Toolkit includes WebSphere Studio extensions.

The Multimodal Tools package includes:

- Multimodal Toolkit V4.3.1 for WebSphere Studio V5.1.1

Or

- Multimodal Toolkit V4.3.2 for WebSphere Studio V5.1.2
- Multimodal Browser V4.1.2 (developed in strategic relationship with Opera Software ASA and ACCESS Systems Company), offering the voice-enabled Opera Browser by Opera Software ASA and NetFront Browser by ACCESS Systems.

7. How does XHTML+Voice differ from VoiceXML?

VoiceXML is for voice-only interaction (e.g. voice portals, voice-enabled call centers), and XHTML is for visual interaction or graphical Web content. XHTML+Voice brings the two together to enable Web sites that support both graphical content and spoken interaction for multimodal applications.

8. How does XHTML+Voice work?

XHTML+Voice defines a way to speech-enable new and existing Web applications. It uses mechanisms similar to what a developer is already accustomed to using (i.e. XHTML). This is done by specifying a subset of VoiceXML and its integration into traditional markup applications. In essence, the Web developer can simply associate VoiceXML with specific parts of the application. When the user interacts using voice, the input is returned to the application as if it had been typed or selected. This means that whether a user enters information by voice or using keystrokes in a multimodal application, the program automatically recognizes it as input and handles it accordingly. The developer will not have to write separate programs catering to various types of input.

9. How can XHTML+Voice help developers? What applications and devices are we talking about?

Developers can add voice interaction to graphical Web content for multimodal deployment without having to learn a whole new language.

Through the use of prepackaged dialogs and reusable VoiceXML code, developers can add simple voice dialogs rather than having to learn a new, specialized skill.

Complex voice dialogs can be built by designers well-versed in speech interfaces; such complex dialogs can be re-used by XHTML authors whose expertise lies in traditional Web design.

Web sites enhanced with voice interaction are likely to be of significant advantage when accessing the Web from mobile hand-held devices with small displays and no keyboard. This technology can enable developers to make their content accessible to the mobile device users.

Devices might include mobile phones, personal digital assistants (PDAs), pagers, car navigation systems, mobile game machines, digital book readers, and smart watches.

10. What is XHTML?

The eXtensible HyperText Markup Language (XHTML) is an XML-based markup language for creating visual applications that users can access from their desktops or wireless devices. XHTML is the next generation of HTML 4.01 in XML.

If you have existing programs with HTML pages, you will have to make some structural changes to comply with XHTML conventions. XHTML has replaced HTML as the supported language by the W3C, so future-proofing your Web pages by using XHTML will not only help you with multimodal applications, but will ensure that users with all types of devices will be able to access your pages correctly. For more information, see the XHTML 1.0 specification (using the XHTML 1.0 - Transitional DTD): http://www.w3.org/TR/xhtml1/

11. What is VoiceXML?

The Voice eXtensible Markup Language (VoiceXML) is an XML-based markup language for creating distributed voice applications, just as HTML is a language for distributed visual applications. VoiceXML was defined and promoted by an industry forum, the VoiceXML Forum™, founded by AT&T®, Lucent®, Motorola®, and IBM, and supported by approximately 500 member companies. Updates to VoiceXML are a product of the W3C voice working group. The language is designed to create audio dialogs that feature text-to-speech, pre-recorded audio, recognition of both spoken and DTMF key input, recording of spoken input, telephony, and mixed-initiative conversations. Its goal is to provide voice access and interactive voice response (such as by telephone, PDA, or desktop) to Web-based content and applications.

Users interact with these Web-based voice applications by speaking or by pressing telephone keys rather than through a graphical user interface.

For more information on VoiceXML 2.0, see the Web site: http://www.w3.org/TR/voicexml20/

12. How does XHTML+Voice differ from SALT?

While the SALT and XHTML+Voice specifications deal with multimodal Web applications, XHTML+Voice has the benefit of using markup languages that are already supported by the W3C. XHTML+Voice can enable mainstream Web developers to add speech input and output to Web content to create multimodal applications.

13. Would the SALT Forum members be encouraged to use the XHTML+Voice specification?

Yes. Many members of the SALT Forum are also members of the voice browser working group that has developed VoiceXML. SALT Forum members are also active in the W3C. We hope that they view XHTML+Voice as a means for leveraging work that has already been performed.

14. What is the Multimodal Browser?

The Multimodal Tools package includes two browsers, developed in a strategic relationship with Opera Software (based on the Opera Browser V7.55) and ACCESS Systems Company (based on the NetFront Browser V3.1 by ACCESS Systems). They are each enhanced with extensions that include the IBM speech recognition and text-to-speech technology, allowing you to view and interact with multimodal applications that you have built using XHTML+Voice.

When you install the Multimodal Browsers, the icons for the browsers appear on your desktop, and you can use them to open the browsers and run your multimodal applications.

For more information on the IBM/Opera strategic relationship, go to:
http://www.opera.com/products/verticals/multimodal/index.dml

For more information on the IBM/ACCESS relationship, go to:
http://www.access-us-inc.com/Prod_NetFront_nf_xhtml.html

15. What skills should I have before I start developing multimodal applications?

At a minimum, you should have experience with Web development (HTML) and Web servers. In addition, experience with XML (or better yet, VoiceXML) and XHTML can make you more proficient at developing multimodal applications.

16. What are the prerequisites for the Multimodal Tools installation?

The **mm_readme.html** (available in the download package) lists the exact hardware and software requirements for installation. The most important software requirement is that you must have a WebSphere Studio base product (obtained separately) installed on your system before installing the Multimodal Tools. The Multimodal Tools can be installed only on systems with:

- **Microsoft® Windows® 2000 system** (Service Pack 2 or higher)
- Plus an installed version of either:

  WebSphere Studio Site Developer V5.1.1 or **WebSphere Studio Application Developer V5.1.1 for Multimodal Toolkit V4.3.1** included in **Multimodal Tools V4.1.2**.

  Or

  WebSphere Studio Site Developer V5.1.2 or **WebSphere Studio Application Developer V5.1.2 for Multimodal Toolkit V4.3.2** included in **Multimodal Tools V4.1.2.2**.

  Refer to http://www.ibm.com/developerworks/websphere/zones/studio/.

Important:

- You should uninstall **any previous versions** of the Multimodal Toolkit, Multimodal Browser, or WebSphere Voice Server SDK before installing this version of the Multimodal Tools. Applications, projects, and files developed in previous (beta) versions are not supported with this version.
- You will be able to import your current projects into this version.

17. The Multimodal Tools perspective does not appear in the list of perspectives. What should I do?

If you have been using WebSphere Studio, you will have to use Update Manager to confirm and configure updated plugins for WebSphere Studio. After you install the Multimodal Toolkit, when you open WebSphere Studio, a dialog will ask you to configure the updates before the Multimodal Tools feature will be visible in the Studio.

- If the **Update Manager** pop-up window appears after you install the toolkit, select **Yes** to open Update Manager, and accept all changes by clicking the top-level check box. Click **Finish** and restart WebSphere Studio.
- If you originally selected No (or if you want to check for other updates), you can manually select **Help > Software Updates > Pending Changes**, click all the check boxes, and then press the **Finish** button. You can periodically check Update Manager for new additions to WebSphere Studio.

18. How do I write a multimodal application using the Multimodal Toolkit?

After installing the Multimodal Toolkit (and the other programs included in the download), you can read the **Getting Started Guide** (getting_started.pdf), which provides guided practice in developing a basic multimodal application using the Multimodal Toolkit. You can locate the guide using the Installation panel and from the Help menu in WebSphere Studio (Help > Help Contents > Multimodal Tools > Getting started). In addition, you can run sample multimodal applications included in the sample folder where you installed the Multimodal Browser (by default, Program Files\IBM\Multimodal Browser\sample).

19. How is this release different from previous versions?

The Multimodal Tools in this release include performance enhancements and increased functionality for creating and testing multimodal applications. There are several important differences from previous versions:

- New **XHTML+Voice Programmer's Guide** (xvguide.pdf) providing information on elements and attributes in X+V, as well as sample applications (see online help: Help > Help Contents > Multimodal developer information > Related documents).
- Integrated **Embedded ViaVoice** speech engines which include enhanced recognition rates and larger grammar support than in previous versions. Included in the toolkit and the multimodal browser is the support for pre-compiled grammars. See programmer's guide for further details.
- New wizard for **generating the <sync> tag** (see the online help: X+V editor > Tasks > Editing X+V files > Connecting XHTML to the VoiceXML).
- New wizard for **generating a grammar** from an HTML control (see the online help: X+V editor > Tasks > Editing X+V files > Adding a grammar in the X+V file).
- Support for the **Grammar Test Tool** using the right-click option **Test Grammar**, which lets you test a grammar with enumeration, speech, and text (see the online help: Grammar test tool >

Tasks).

- Support for Aural Cascading Style Sheets (ACSS), currently only supported on the Opera browser.

- Extended Voice Preferences provides more voice preferences for customized TTS and other features.

- Support for Speech Recognition Grammar Specification (SRGS) XML and ABNF formats.

- Speech Synthesis Markup Language (SSML) support.

- Added support for the VoiceXML property <property name="maxspeechtimeout">.

- Recognition result confidence score has improved granularity.

- Added support for the <grammar expr="..."> attribute.

20. Are any sample multimodal applications included?

The toolkit includes sample applications that you can open in the Multimodal Browser. For example, the pizza sample has a voice-enabled ordering form. The transaction sample shows a voice-enabled billing address form (the page is similar to the scenario that you can develop using the *Getting Started Guide*). The **sample** folder is located in the Multimodal Browser installation directory. To view the samples, do one of the following:

- Open the Multimodal Browser, and click **File > Open**, and use the **Browse** button to locate the sample directory (by default, Program Files/IBM/Multimodal Browser/sample). Expand the folders, and double-click the sample file (.mxml or .html).
- Open a Multimodal Browser, and from Windows Explorer, drag the sample file (.mxml or .html) into the browser.

Note: When opening an .mxml file using **File > Open**, change the "Files of type" field to show **All files**.

21. What key do I use for the Push-to-Talk (or microphone) button?

When the application opens, usually a voice prompt begins immediately. If it doesn't, click in a field to hear the voice prompt. Press the **Scroll Lock** key on the keyboard to activate the microphone, listen for the tone, pause briefly to let the microphone engage, and talk into your microphone. Pause briefly before releasing the **Scroll Lock** key when you finish.

You can change the listening mode and keyboard Push-to-Talk button (as well as the log level).

- In the Opera Browser, select **Tools > Preferences > Voice**, and use the settings. For example, from the **Key to talk** drop-down list, select **Insert**. When you re-start the browser, use the **Insert** key as the Push-to-Talk button.
- In the NetFront Browser by ACCESS Systems, select **File > Preferences** > **Voice**, and use the settings. For example, from the **PTT Key** drop-down list, select **Insert**. When you re-start the browser, use the **Insert** key as the Push-to-Talk button.

New in this release is the ability to use Command, control, and Content Navigation. By selecting this option, you can use global command words to activate controls in the browser, instead of the grammars in the X+V applications.

In addition, each browser has three listening (or talk key) modes, which you can select **Voice** preferences.

- In **Push-to-talk** (or for the Opera browser, **Hold key while talking**) mode, you press and hold the button on the device while speaking, and then release the button (default selection).
- In **Push-to-activate** (or **Press key, then talk**) mode, press and release the button, and then talk. When you finish speaking, it detects silence and automatically stops listening (if there is background noise, it might take a moment for the system to detect the end of speech).

  Note: When using the VoiceXML `<record>` tag, the **Push-to-activate** mode has a slightly different behavior. Press and release the button, begin speaking, and then push and release the button again to signal the end of the response.
- In **Auto-push-to-activate** (or **Key not required to talk**) mode, the browser automatically sounds a tone when it is ready to record your response. When you finish speaking, the device detects silence and automatically stops listening (if there is background noise, it might take a moment for the device to detect the end of speech).

22. What are the parts of a multimodal application?

A multimodal application consists of two main components: the visual component, written with XHTML, and voice component, written in VoiceXML. The visual and voice components are linked together using an event handler or the new <sync> tag.

An event handler specifies an action to be performed when a particular event (such as a mouse click) takes place. In XHTML+Voice, event handlers enable interaction between XHTML and VoiceXML markup.

XHTML+Voice supports the event types defined in HTML 4.01. These event types have been translated into XML Events event types, thereby removing the "on" prefix. XHTML+Voice also supports the following VoiceXML 2.0 event types: nomatch, noinput, error, and help. In addition, XHTML+Voice supports the vxmldone event, which is generated when a voice handler completes.

Here is an example of using XML Events in your XHTML+Voice application:

```
<input type="text"
     id="pizzaQuantity"
     ev:event="focus"
     ev:handler="#voice_quantity"/>
```

In the above example, whenever the user clicks on the <input> element, a "focus" event is generated. This causes the VoiceXML form with ID "voice_quantity" to be activated.

Alternatively, the XHTML+Voice <sync> element adds support for synchronization of data entered using either speech or visual input, as shown in the following example.

```
    <xv:sync xv:input="visual_field_name"
             xv:field="#voice_field_id"/>
```

In the example above, the <sync> tag, added to the VoiceXML in the <head>, connects a visual input defined in the XHTML to the voice field defined in the VoiceXML.

23. How do I test my multimodal file in the Multimodal Browser?

If you developed your application using the Multimodal Toolkit, then after you complete the multimodal file (with file extension .mxml) and grammar files, you can right-click the .mxml file in the navigation pane or in the open .mxml file in the editor, and select **Launch Multimodal Browser**.

Multimodal projects are located in **workspace** directory (which you selected during startup). The .mxml file and any grammar and pool files that you create should be located in the **Web Contents** folder of the project.

Alternatively, you can open the browser and use **File > Open** to locate and open the .mxml file. Remember to change the "Files of type" field to show **All files**.

24. I want to create or edit pronunciations. How do I do this?

The Multimodal Toolkit provides a grammar editor in which you can create JSGF, SRGS XML/ABNF grammar files. The feature automatically creates the phonologies needed by the speech engines. In addition, the toolkit provides a Pronunciation Builder tool that lets you compose and edit pronunciations in the grammar files to add to pronunciation pool files using a simple dialog. For more information on grammar formats and pronunciation building, see the X+V Programmer’s Guide included with the Multimodal Toolkit.

25. What content types are supported for XHTML+Voice files?

The following table shows valid extensions and their corresponding MIME Content types.

Extension	**MIME Content type**
.mxml, .jsm	application/xhtml+voice+xml
.jsgf	application/x-jsgf
.grxml (SRGS XML)	application/srgs+xml
.gram, .abnf (SRGS ABNF)	application/srgs
(Precompiled grammar)	application/x-ibmvocabset
.mxml*	text/html
* We recommend that you add the .mxml extension to the text/html content type so that you can display the XHTML content of an XHTML+Voice document in a browser that does not supportX+V.	

Note that XHTML+Voice files should include the following lines:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//VoiceXML Forum//DTD XHTML+Voice 1.2//EN"
"http://www.voicexml.org/specs/multimodal/x+v/12/dtd/xhtml+voice12.dtd">
<html xmlns=http://www.w3.org/1999/xhtml
  xmlns:ev=http://www.w3.org/2001/xml-events
  xmlns:vxml=http://www.w3.org/2001/vxml
  xmlns:xv=http://www.voicexml.org/2002/xhtml+voice
  xml:lang="en-US">
```

26. Can I install the Multimodal Tools on a Microsoft Windows XP platform?

The Windows 2000 platform is the supported platform for this Multimodal Tools release. The Multimodal Tools have had a limited amount of testing on the Windows XP platform and the installation and running of this release on that platform will not be prevented.

27. What are the known problems and limitations?

The following items are restrictions on ABNF/XML SRGS support:

- Special Rule:$GARBAGE will be treated as an error.
- N-Gram documents/grammars are not supported. If N-Gram documents/grammars are used, the behavior is undefined.
- We do not support a language reference within a rule. Language references of this type will be ignored.
- Weights for alternatives will be discarded by the grammar compiler.
- Repeat Probabilities will be discarded by the grammar compiler.
- Multi-lingual grammars are not supported, except where US English words are used as part of a language where there is no equivalent. For example, IBM has no translation. The word is spelled and pronounced the same regardless of language. If mixed languages are used within a grammar, the behavior is undefined.
- Character encodings are supported: Codepage 1252 for single byte languages or UCS-2 Unicode for multi-byte languages.

The following items are known problems on SSML:

- The <prosody pitch> with the value of "x-high" will sound like "x-low". The workaround is to use a number followed by "Hz" or a relative change instead.
- The <prosody range> with the value of "x-high", "high", "low" or "x-low" will sound like "medium". The workaround is to use a number followed by "Hz" or a relative change instead.

The following items are known problems on Multimodal Tools 4.1.2.2:

- Attributes that have namespace prefixes, such as xv:id, will be erroneously underlined in red in the X+V editor. This can be corrected by installing the fix pack available from the Multimodal Download Web site: http://www.ibm.com/pvc/multimodal. Follow the links to download the Multimodal Tools 4.1.2.2.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.