![Dialogic logo]

Dialogic® PowerMedia™ XMS VoiceXML

Reference Guide

www.dialogic.com

Copyright and Legal Notice

Table of Contents

Revision History

Revision	Release Date	Notes
05-2710-011	April 2017	Updates to support PowerMedia XMS Release 3.3. Call Progress Analysis: Added details on CPA profiles.
05-2710-010	November 2016	Updates to support PowerMedia XMS Release 3.2. Outbound Calling: Updated the section with Global Unique Call ID support. <dlgc:recordcall>: Added new section. <grammar>: Added note to tag-format attribute and updated the Details section.
05-2710-009 (Updated)	May 2016	Video Playback and Record: Added how to use the input video stream's resolution and framerate as the target encoding resolution and framerate.
05-2710-009	March 2016	Updates to support PowerMedia XMS Release 3.1. Outbound Calling: Added new section.
05-2710-008 (Updated)	October 2015	Connection Protocol Subobjects: Updated media and media.format. <field>: Updated the type attribute.
05-2710-008	September 2015	Updated for PowerMedia XMS Release 3.0. <audio>: Updated the type attribute. <media>: Updated the type attribute. <par>: Updated the Details and Example sections. <record>: Updated the type attribute.
05-2710-007 (Updated)	June 2015	<speak>: Added details on SSML fragments. Added details on when the <speak> element is allowed. Added <media> as a parent. Added details to the example. Call Progress Analysis: Added details on enabling the CPA feature.
05-2710-007 (Updated)	May 2015	Call Progress Analysis: Added com.dlgc.cpa.signals property table specifying valid values. <record>: Added Details section with information on dlgc:dest and dlgc:destexpr.

Revision	Release Date	Notes
05-2710-007	February 2015	Updates to support PowerMedia XMS Release 2.4. Audio Playback Control: Added details on enabling dlgc domain name. Video Playback and Record: Added new section. <dlgc:file>: Added new section. <par>: Added new section. <record>: Added dlgc:videoname and dlgc:videotype attributes. Added details on enabling dlgc domain name.
05-2710-006	October 2014	Updates to support PowerMedia XMS Release 2.3. Call Progress Analysis: Added new section. Restrictions and Limitations: Added new section. <audio>: Updated playcontrol attribute to dlgc:playcontrol. <object>: Updated Example section. <record>: Added dlgc:dest, dlgc:destexpr, and dlgc:lang attributes. <say-as>: Updated Property and Example sections with com.dlgc.sayas.
05-2710-005	June 2014	Session Variables: Added note to requesturi in Connection Protocol Subobjects section. Behavior Changes: Added new section. <audio>: Added that VoiceXML is capable of generating DTMF with the <audio> element.
05-2710-004	March 2014	Updates to support PowerMedia XMS Release 2.2. Audio Playback Control: Added new section. <say-as>: Updated section with support for builtin engine.
05-2710-003	November 2013	Appendix: VoiceXML Elements and Attributes: Added new section.
05-2710-002	October 2013	Updates to support PowerMedia XMS Release 2.1. Overview of VoiceXML: Added new section.
05-2710-001	February 2013	Updates to support PowerMedia XMS Release 2.0.
05-2710-001-01	October 2012	Initial release of this document.

Revision	Release Date	Notes
Last modified: April 2017		

Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

1. Welcome

This Reference Guide provides a Voice Extensible Markup Language (also referred to herein as "VoiceXML" or "VXML") dictionary for users of the Dialogic® PowerMedia™ Extended Media Server (also referred to herein as "PowerMedia XMS" or "XMS"). It contains an alphabetical reference of supported VoiceXML elements and provides information about application properties, SSML support, session variables, and application variables.

For installation and configuration instructions, refer to the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide*.

2.   Overview of VoiceXML

VoiceXML for PowerMedia XMS is designed for creating audio dialogs that feature synthesized speech, digitized audio, speech recognition, DTMF key input, speech recording, telephony, and mixed initiative conversations.

VoiceXML is a W3C standard scripting language for playing text to speech and audio prompts, and for collecting DTMF and voice input. ECMAScript, also known as JavaScript, is a programming language adopted by the European Computer Manufacturer's Association as a standard for performing computations in Web applications. ECMAScript is the official client-side scripting language of VoiceXML. ECMAScript is a limited programming model for simple data manipulation. The VoiceXML language described by W3C is used to create plain text pages using specific XML-based language.

The interpreter executes VoiceXML dialogs on an RTP stream. Each dialog represents an announcement, menu, or other script. The dialogs finish when they have posted information to a web server or returned a namelist back to the command that invoked the browser. If additional dialogs are needed, then the application runs another script.

The PowerMedia XMS VoiceXML service provides interactive dialogs with a remote caller by interpreting VoiceXML pages. The dialogs are SIP-based, as described in RFC 5552 (SIP Interface to VoiceXML Media Services).

VoiceXML Interpreter

PowerMedia XMS includes interpreters for VoiceXML 2.0 and VoiceXML 2.1.

If the PowerMedia XMS receives a SIP INVITE request directed to the dialog service, a VoiceXML session begins. The VoiceXML session initially gets (fetches) and executes the VoiceXML script that is specified in the SIP Request-URI parameter, voicexml.

Once specified, a VoiceXML script URI remains in effect until it completes execution or the session is stopped.

For installation and configuration instructions, refer to the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide*.

3. VoiceXML Application Reference

Application Properties

VoiceXML application properties allow you to set platform defaults for a session and alter platform behavior when the application executes. Unsupported properties are allowed but ignored at execution. Properties follow the standard VoiceXML scoping rules.

Usage

To use a property in a VoiceXML application, specify the value attribute for the `<property>` tag. For example, suppose you want to have an application that takes advantage of a fast caching policy, so that your application (VoiceXML documents, grammars, scripts, and so forth) load quickly. However, during development you need your application to use a safe caching policy so that you can develop and debug. During development, set the following properties in the application root:

```
<vxml version="2.1">
    <property name="documentmaxage" value="0"/>
    <property name="audiomaxage" value="0"/>
    <property name="grammarmaxage" value="0"/>
    <property name="scriptmaxage" value="0"/>
...
</vxml>
```

When you deploy the application, either change the property values from 0 to non-zero values, or remove the properties.

Note: A property is scoped to the level where it is specified. For example, a property specified at the application level provides the defaults for all loaded documents. A property at the form level overrides properties set previously at the application or document level.

Supported Properties

The VoiceXML Interpreter supports the following types of properties:

- DTMF
- Fetch
- Prompt
- Recognition

DTMF Properties

The following table describes the VoiceXML Interpreter's DTMF properties:

Property	Data Type	Default	Description
interdigittimeout	Number	3.0s	Timeout value between DTMF occurrences.
termchar	0-9 # * ""	#	Terminating DTMF character for DTMF input recognition. Using the blank option ("") allows you to specify all DTMF characters as input items, including the pound key (#).
termtimeout	Number	0s	Termination timeout for DTMF occurrences.

Fetch Properties

The following table describes the VoiceXML Interpreter's fetch properties.

Property	Data Type	Default	Description
audiofetchhint	{safe | prefetch}	safe	Currently not supported.
audiomaxage	Number	N/A	Maximum acceptable age of cached audio resources, in seconds.
audiomaxstale	Number	N/A	Maximum acceptable staleness of expired, cached audio resources, in ms. Specifying an audiomaxstale value lets a VoiceXML document use audio resources contained in the cache for a given period after the resources have expired.
datafetchhint	{safe | prefetch}	safe	Currently not supported.
datamaxage	Number	N/A	Maximum acceptable age of cached XML data, in seconds.
datamaxstale	Number	N/A	Maximum acceptable staleness of expired cached XML documents, in seconds. Specifying a datamaxstale value lets a VoiceXML document use XML data contained in the cache for a given period after the data has expired.

Property	Data Type	Default	Description
documentfetchhint	{safe | prefetch}	safe	Currently not supported.
documentmaxage	Number	N/A	Maximum acceptable age of cached documents, in seconds.
documentmaxstale	Number	N/A	Maximum acceptable staleness of expired cached documents, in seconds. Specifying a documentmaxstale value lets a VoiceXML document use VoiceXML documents contained in the cache, for a given period after the documents have expired.
fetchaudio	Number	N/A	URI of the audio to play while waiting for a document to be fetched.
fetchaudiodelay	Number	0s	Amount of time to wait at the start of a fetch delay before playing the fetchaudio source. This is useful if a fetch delay is short, when it may be preferable not to start playing fetchaudio that will be cut-off almost immediately.
fetchaudiominimum	Number	2s	Minimum time interval to play a fetchaudio source, once started, even if the fetch result arrives in the meantime. This is useful when you do not want the fetchaudio heard by a user to be stopped too quickly.
fetchtimeout	Number	10s	Amount of time to wait before a timeout event is raised on HTTP fetches.
grammarfetchhint	{safe | prefetch}	safe	Currently not supported.
grammarmaxage	Number	N/A	Maximum acceptable age of cached grammars, in seconds.

Property	Data Type	Default	Description
grammarmaxstale	Number	N/A	Maximum acceptable staleness of expired cached grammars, in seconds. Specifying a grammarmaxstale value lets a VoiceXML document use grammar resources contained in the cache, for a given period after the resources have expired.
objectfetchint	{safe | prefetch}	safe	Currently not supported.
objectmaxage	Number	N/A	Maximum acceptable age of cached objects, in seconds.
objectmaxstale	Number	N/A	Maximum acceptable staleness of expired cached objects, in seconds. Specifying an objectmaxstale value lets a VoiceXML document use objects contained in the cache for a given period after the objects have expired.
scriptfetchhint	{safe | prefetch}	safe	Currently not supported.
scriptmaxage	Number	N/A	Maximum acceptable age of cached scripts, in seconds.
scriptmaxstale	Number	N/A	Maximum acceptable staleness of expired cached scripts, in seconds. Specifying a scriptmaxstale value lets a VoiceXML document use script resources contained in the cache for a given period after the resources have expired.

Prompt Properties

The following table lists the VoiceXML Interpreter's prompt properties:

Property	Data Type	Default	Description
bargein	boolean	true	Allows or disallows prompt bargein.
bargeintype	{speech | hotword}	speech	Type of barge-in performed in response to voice input.
timeout	Number	3.4s	Elapsed silent time before a noinput event is raised.

Recognition Properties

The following table lists the VoiceXML Interpreter's recognition properties:

Property	Data Type	Default	Description
completetimeout	0.2s - 10s	0.8s	Length of silence after speech before a result is either accepted or rejected with a nomatch event. This value applies when the speech prior to silence is a complete match of an active grammar.
confidencelevel	0.0 - 1.0	0.5	Threshold of the speech recognition confidence level. The VoiceXML Interpreter rejects user inputs if the confidence level is below this threshold. Valid values range from 0.0 to 1.0.
incompletetimeout	0.2s - 10s	1s	Length of silence after speech before a result is finalized. This value applies when the speech prior to silence is either: • An incomplete match of all active grammars, or • A complete match of an active grammar, but the user can speak additional grammar items. In the first case, when the timeout is triggered, the VoiceXML Interpreter rejects the partial result and raises a nomatch event. In the second case, the VoiceXML Interpreter accepts the speech.

Property	Data Type	Default	Description
inputmodes	{dtmf | dtmf voice | voice}	dtmf voice	Space separated list defining the type of inputs allowed. Valid values: • dtmf = Speech recognition is disabled. • voice = DTMF recognition is disabled.
maxspeechtimeout	0s - 600s	15s	Maximum duration of user speech. In VoiceXML 2.0, if this time elapses before the user stops speaking, a maxspeechtimeout event is raised. Use 0 for no limit.
maxnbest	1 - 500	1	Maximum number of results returned by the recognizer. Also represents the maximum size of the application.lastresult$[*i*] array.
recordutterance	boolean	false	Indicates whether to enable or disable utterance recording during recognition. • true: Enables utterance recording during recognition. • false: Disables utterance recording. When utterance recording is enabled, the following variables contain recording values: • lastresult$.recording contains the user's utterance. • lastresult$.recordingsize contains the recording size, in bytes. • lastresult$.recordingduration contains, the recording duration in ms when a recognition result is returned. If a form item is filled, the corresponding form item shadow variables also are set to these values.
recordutterancetype	{audio/x-wav}	audio/x-wav	Media type to use for utterances recorded during recognition.

Property	Data Type	Default	Description
sensitivity	0.0 - 1.0	0.5	Sensitivity level to input. Valid values range from 0.0 (least sensitive) to 1.0 (most sensitive).
speedvsaccuracy	0.0 - 1.0	0.5	Provides information to the recognizer of the desired balance between speed and accuracy. Values range from 0.0 (fastest recognition) to 1.0 (highest accuracy).
universals	{cancel | exit | help | none}	none	Space-separated list of the universal commands to activate. For example: `"help" "cancel" "exit"`

SSML Support

VoiceXML for PowerMedia XMS will follow the Speech Server Markup Language (SSML) 1.0 standard for speech control interface. It will have its own SSML processor to interpret speech tags and provide to VoiceXML application a standard way to control aspects of speech such as pronunciation, volume, pitch, and rate.

PowerMedia XMS will only support a subset of the SSML 1.0 standard. Although all elements will be parsed and processed, the only SSML element to be integrated with the PowerMedia XMS media engine is the <audio> element. It specifies the audio files to be played. For TTS enabled application, all SSML commands will be transferred as-is to the speech server via the Media Resource Control Protocol (MRCP) interface. Refer to the speech vendor specification for the SSML support coverage.

Session Variables

VoiceXML session variables are read-only variables that hold information about the platform during an entire user session. At the beginning of a user session, the VoiceXML Interpreter sets and declares the session variables. These variables follow the usual VoiceXML scoping rules and are at the highest tier of the scope chain.

As PowerMedia XMS is on top of the SIP stack, VoiceXML supported session variables are defined and specified in RFC 5552.

Usage

Session variables are referenced in any VoiceXML document during a call session.

<block>

   <prompt>
      The Caller ID is <value expr="session.connection.remote.uri"/>
   </prompt>
...
</block>

The following table lists the available session variables:

Session Variable	Description
session.connection.aai	Application-to-application information passed during connection setup. For example, this information can be set by the <transfer> element.
session.connection.initialuri (dialogic)	URI of the first page of VoiceXML used in the call.
session.connection.local.uri	URI that addresses the local VoiceXML Interpreter device. For a SIP inbound call, this is the value of the INVITE To: header.
session.connection.originator (dialogic)	Directly references the local or remote URI. For example, the following ECMAScript returns true if the remote party initiated the connection: `var caller_initiate = session.connection.originator == session.connection.remote.uri.`
session.connection.protocol.name	Name of the connection protocol. This name can also represent the subobject name for protocol-specific information, as in the following example: `session.connection.protocol[session.connection. protocol.name].*`
session.connection.protocol[session. connection.protocol.name].*	These subobject variables provide protocol specific information. For example, if session.connection.protocol.name is 'sip', then session.connection.protocol.sip.media would specify the media informations. For more information, see Connection Protocol Subobjects.
session.connection.protocol.version	Connection protocol version.

Session Variable	Description
session.connection.redirect	An array representing the connection redirection paths. The first element is the original called number and the last element is the last redirected number. Each element of the array contains a URI, PI (presentation information), SI (screening information), and reason property. Valid values for the reason property are: • deflection during alerting • deflection immediate response • mobile subscriber not reachable • no reply • unknown • user busy
session.connection.remote.uri	URI that addresses the remote device. For a SIP inbound call, this is the value of the INVITE From: header. For a PSTN inbound call, this is the calling party number, regardless of the Presentation Restricted setting.

Connection Protocol Subobjects

The session.connection.protocol session variable contains subobject variables that provide protocol-specific information. For example, if session.connection.protocol.name is 'sip', then session.connection.protocol.sip.media gives the media information.

The following table describes the connection subobjects available for the SIP protocol when the VoiceXML Interpreter is used with the call server. These subobjects are available within the session.connection.protocol.sip object.

Subobject Name	Description
requesturi	Request-URI from the SIP INVITE. **Note:** URI parameter names are converted to lowercase when creating session variables (as described in RFC 5552).
requesturi.voicexml	VoiceXML parameter from the Request-URI. This information is also available through the session.connection.initialuri variable.
requesturi.aai	Application-to-application information from the Request-URI. Used to specify a JSON value (as described in RFC 4627). This information is also available through the session.connection.aai variable.

Subobject Name	Description
requesturi.ccxml	Ccxml information from the Request-URI. Used to specify a JSON value (as described in RFC 4627) that is mapped to the session.connection.ccxml VoiceXML session variable.
requesturi.maxage	Used to set the max-age value of the Cache-Control header in conjunction with VoiceXML documents fetched using HTTP (as described in RFC 2616). If omitted, the VoiceXML Media Server will use a default value.
requesturi.maxstale	Used to set the max-stale value of the Cache-Control header in conjunction with VoiceXML documents fetched using HTTP (as described in RFC 2616). If omitted, the VoiceXML Media Server will use a default value.
requesturi.method	Used to set the HTTP method applied in the fetch of the initial VoiceXML document. Allowed values are "get" or "post" (case-insensitive). Default is "get".
requesturi.postbody	Used to set the application/x-www-form-urlencoded encoded (HTML4) HTTP body for post requests (or is otherwise ignored).
media	Contains information about media (from SDP). The variable is "undefined" until the line is connected. Prior to being connected, any access to a child or an array element will throw a semantic error.
media.type	This required property indicates the type of the media associated with the stream. The value is a string. It is strongly recommended that the following values are used for common types of media: "audio" for audio media.
media.direction	This required property indicates the directionality of the media relative to session.connection.originator. Defined values are sendrecv, sendonly, recvonly, and inactive.
media.format	This property is optional and will return "undefined".

Subobject Name	Description
headers	Named arrays containing the SIP headers. The header names are lower case as specified in RFC 5552.
	For instance:
	session.connection.protocol.sip.headers["contact"] refers to the Contact SIP header content.
	Note: The headers.name syntax is also valid only if the header name is conforming to ECMAScript variable definition.
	For instance:
	session.connection.protocol.sip.headers.contact is correct but session.connection.protocol.sip.headers.call-id is not due to the use of the "-" symbol.

Example

The following shows the variables used when a SIP call is in session:

```
session.connection.protocol.name="sip"
session.connection.protocol.version="2"
session.connection.protocol.sip.requesturi="sip:dialog@mediaserver.example.net;
    voicexml=http://vxmlserver.example.net/cgi-bin/script.vxml;aai=information"
```

Application Variables

The read-only application.lastresult$ variable is an array of elements containing information about the last recognition to occur within an application. Each array element represents a possible recognition result.

Recognition results are sorted by the confidence score of the recognition, from highest to lowest. Using application.lastresult.$ without an index is the same as referencing the first element of the array, application.lastresult$[0].

The following table describes the subobjects available for each possible recognition result:

Name	Description
application.lastresult$[$i$].bargeintime	Duration that elapsed until the user gave input.
application.lastresult$[$i$].confidence	Whole utterance confidence level for this interpretation. Valid values range from 0.0 (minimum) through 1.0 (maximum).
application.lastresult$[$i$].inputmode	Indicates whether the user used voice or DTMF to input this result.
application.lastresult$[$i$].interpretation	Semantic interpretation of the user's input.

Name	Description
application.lastresult$[*i*].markname	Name of the last `<mark>` executed before the user gave input, or before the end of playback occurred.
application.lastresult$[*i*].marktime	Number of milliseconds between when the last `<mark>` was executed and the user gave input, or the end of playback occurred.
application.lastresult$[*i*].recording	Reference to the user's utterance, if the recordutterance property is set.
application.lastresult$[*i*].recordingduration	Duration of the recording of the user's utterance, in ms.
application.lastresult$[*i*].recordingsize	Size of the recording of the user's utterance in bytes.
application.lastresult$[*i*].utterance	Raw string of words recognized for this interpretation. For a DTMF grammar, this variable contains the string representation of the matched digits.

Application variables are referenced in the same scope as any recognition that occurred during a call session.

Most of the application variables are copied into shadow variables as subobject of the name$ of the field. See the Shadow Variables section for details.

Example

```
<field>
    <prompt>Say yes or no</prompt>
    <filled>
        <if cond="application.lastresult$.confidence &lt; 0.5">
            <goto nextitem="confirm"/>
        <else>
            <goto next="next menu.html"/>
        </if>
    </filled>
</field>
```

Shadow Variables

The shadow variable is subobject of the main elements name like record, transfer, and field.

For instance, a record size (in bytes) can be obtained by evaluating the ECMAScript expression:

MyRecord.size, where MyRecord is the name$ of the record field.

\<record\> Shadow Variables

Name	Description
name$.duration	The duration of the recording in milliseconds.
name$.size	The size of the recording in bytes.
name$.termchar	If the dtmfterm attribute is true, and the user terminates the recording by pressing a DTMF key, then this shadow variable is the key pressed (e.g., #). Otherwise it is undefined.
name$.maxtime	Boolean, true if the recording was terminated because the maxtime duration was reached.
name$.recording	The variable that stores a reference to the recording, or undefined if no audio is collected. Like the input item variable associated with a \<record\> element (as described in Section 2.3.6 of VoiceXML Specification), the implementation of this variable may vary between platforms.
name$.recordingsize	The size of the recording in bytes, or undefined if no audio is collected.
name$.recordingduration	The duration of the recording in milliseconds, or undefined if no audio is collected.

\<transfer\> Shadow Variables

Name	Description
name$.duration	The duration of a call transfer in seconds. The duration is 0 if a call attempt was terminated by the caller (using a voice or DTMF command) before the outgoing call begins.
name$.inputmode	The input mode of the terminating command (dtmf or voice), or undefined if the transfer was not terminated by a grammar match.
name$.utterance	The utterance text used if transfer was terminated by speech recognition input or the DTMF result if the transfer was terminated by DTMF input; otherwise it is undefined.

\<field\> Shadow Variables

Name	Description
name$.utterance	The raw string of words that were recognized. The exact tokenization and spelling is platform-specific (e.g., five hundred thirty or 5 hundred 30 or even 530). In the case of a DTMF grammar, this variable will contain the matched digit string.
name$.inputmode	The mode in which user input was provided: dtmf or voice.
name$.interpretation	An ECMAScript variable containing the interpretation (as described in Section 3.1.5 of VoiceXML Specification).
name$.confidence	The confidence level for the name field and may range from 0.0-1.0. A value of 0.0 indicates minimum confidence, and a value of 1.0 indicates maximum confidence. A platform may use the utterance confidence (the value of application.lastresult$.confidence) as the value of name$.confidence. This distinction between field and utterance level confidence is platform-dependent. More specific interpretation of a confidence value is platform-dependent since its computation is likely to differ between platforms.

\<field\> Shadow Variables Linked to \<mark\> Element

Name	Description
name$markname	The name of the mark last executed by the SSML processor before barge-in occurred or the end of audio playback occurred. If no mark was executed, this variable is undefined.
name$marktime	The number of milliseconds that elapsed since the last mark was executed by the SSML processor until barge-in occurred or the end of audio playback occurred. If no mark was executed, this variable is undefined.

Example

```
<field name="myfield">
    <prompt>Say yes or no</prompt>
    <filled>
        <if cond="myfield.confidence &lt; 0.5">
            <goto nextitem="confirm"/>
        <else>
            <goto next="next_menu.vxml"/>
        </if>
    </filled>
</field>
```

Audio Playback Control

The audio playback control feature allows the user to control audio playback by entering specific dtmf digits. This feature is available only for <audio> element that plays a file (not an alternate tts content).

To enable this feature, the optional attribute dlgc:playcontrol must be set to "true" inside the <audio> element.

The following declaration must be used to enable the dlgc domain name:

```
xmlns:dlgc="http://www.dialogic.com/xmlns"
```

<audio> Attribute	Description
dlgc:playcontrol	Enable or disable the playback control for that audio element. Valid values: • true • false (default) Attribute is optional. If not present, playback control is disabled for the audio element.

The dtmf map can be set using the property com.dlgc.playcontrol.

If the property is explicitly set, all desired commands must be defined. If a command is omitted, no dtmf key will be mapped to the related action except for resume which will be mapped to pause. If the skip interval is omitted, the default will be set to 5s.

Property Name	Description
com.dlgc.playcontrol	If the com.dlgc.playcontrol property is not explicitly specified, the default values will be forward=9, backward=7, stop=#, pause=8, resume=8, restart=*, and interval=5s.
	List of space separated commands:
	• forward=[dtmf]
	• backward=[dtmf]
	• pause=[dtmf]
	• resume=[dtmf]
	• restart=[dtmf]
	• stop=[dtmf]
	There is a special command to set the time interval (in seconds) to skip with backward and forward command:
	• interval=[time]s

Example

```
<form>
    <property name="com.dlgc.playcontrol" value="pause=2 backward=1 forward=3 stop=#"/>
    <!— resume is automatically mapped to dtmf 2, skip interval is 5s, there is no restart key -->
    <block>
        <prompt bargein="false">
            <audio src="file://vxml/audio_clip_newscast.wav" playcontrol="true"/>
        </prompt>
    </block>
</form>
```

Note: It is strongly recommended to set the bargein to "false" for the prompts where playcontrol is enabled. Otherwise, a conflict with dtmf recognition grammar could occur.

Video Playback and Record

The video playback and record feature allows the user to perform video playback and record.

To play video, the call must allow video. The property com.dlgc.media.type is used to select media type for the call.

Property Name	Description
com.dlgc.media.type	Select media type for the call. Valid values:
	• audio (default)
	• video
	• audiovideo

If the property changes, VoiceXML should request a re-INVITE to activate/clear the required media.

Once the call is set to manage video stream, the script could play or record audio and video files.

To record a video stream, the <record> field element is used with the dlgc:videoname and dlgc:videotype attributes.

<record> Attribute	Description
dlgc:videoname	Specify the VoiceXML name for the video recording.
dlgc:videotype	Define the video type. Valid values: • video/x-vid • video/3gpp Attribute is optional. If not present, use the API default.

To enable the video recording, the dlgc:videoname attribute must contain the VoiceXML name of the video recording (like the name attribute is for audio recording).

The optional dlgc:videotype attribute can be used to change the PowerMedia XMS API default video type (video/x-vid). The video/x-vid and video/3gpp types are supported.

The input video stream's resolution and framerate can be identified by the system and set as the target encoding resolution and framerate when using H.264, VP8, or VP9 video codecs. To use the input video stream's resolution parameters, set the frame height and frame width to 0. To encode all frames without skipping any, set the framerate to 0.

If the dlgc:dest attribute is used to specify an audio destination, the video destination file will be the basename of the audio URI following with .vid or .3gp according to the dlgc:videotype attribute.

To play a video file, the <par> element must be used. It must include two <media> child elements; the first for the audio part, the second for the video part. The <media> element is similar to the <audio> element. To play the video recording, the dlgc:videoname attribute of <record> element can be referenced in the expr attribute of the video <media> element.

To submit the video, the namelist has to include the VoiceXML video recording name (exactly as for audio recording).

Example

```
<?xml version="1.0" encoding="UTF-8"?>
   <vxml
   xmlns="http://www.w3.org/2001/vxml"
   xmlns:dlgc="http://www.dialogic.com/xmlns" version="2.1">
   <form>
   <property name="com.dlgc.media.type" value="audiovideo"/>
   <record name="myaudio" dlgc:videoname="myvideo"
   dlgc:dest="file://vxml/myrecord.wav" dlgc:videotype="video/x-vid"
   beep="true" maxtime="10s" finalsilence="1000ms" dtmfterm="true">
   <prompt>
   <media src="file://vxml/recording.wav"/>
   </prompt>
   </record>
   <block>
   <log>
   <value expr="myvideo$.duration"/>
   <value expr="myvideo$.size"/>
   <value expr="myvideo$.http"/>
   </log>
   <prompt>
   <par>
   <media expr="myaudio"/>
   <media expr="myvideo"/>
   </par>
   </prompt>
   <submit next="http://mysubmitserver/vxml/submit.php" method="post"
   enctype="multipart/form-data" namelist="myaudio myvideo"/>
   </block>
   </form>
</vxml>
```

Call Progress Analysis

Call Progress Analysis (CPA) detects the type of signal received in the RTP stream. It also detects fax machines needed to switch over to a fax receiver. There are advanced capabilities used to detect answer-machine, voice, fax or any other signals similar to tones.

There are two different kinds of implementation PowerMedia XMS uses to achieve this goal:

- CPA at Session Connection (Incoming or Outgoing)
- Signal after Session is Established

CPA at Session Connection (Incoming or Outgoing)

This feature detects answer-machine, voice, fax or any registered signal at the very beginning of the document execution.

To enable the CPA feature, add the "cpa=yes" parameter in the SIP Request-URI of the INVITE sent to VoiceXML. For example:

```
sip:dialog@myserver;cpa=yes
```

Note: This feature activates low-level detection and could delay the start of the VoiceXML script.

The cpaprofile="xxxx" parameter specifies the name of the CPA profile to be used when PowerMedia XMS performs CPA detection. The cpaprofile parameter can only be specified when "cpa=yes" is specified. CPA profiles can be defined on the WebGUI.

The associated property com.dlgc.cpa.signals filters the signals that will throw the event.

As soon as the SIP connection is established, the incoming stream is analyzed. When the analysis is terminated, the found signal is searched in the property. If present, a special event (com.dlgc.cpa) is thrown inside the VoiceXML Interpreter. This event can be caught by a <catch event="com.dlgc.cpa"> element, and the signal name can be found in the _message variable.

Property Name	Description
com.dlgc.cpa.signals	Valid values: • unknown • answer-machine • voice • fax • custom tone name (e.g., CNG)

Example: CPA

```
<property name="com.dlgc.cpa.signals" value="fax voice"/>
 <catch event="com.dlgc.cpa">
   <log> received cpa info : <value expr="_message"/></log>
   <if cond="_message == 'fax'">
     <goto next="#fax"/>
   <else/>
   <elseif cond="_message == 'voice'"/>
     <goto next="#welcome"/>
   <else/>
     <exit/>
   </if>
 </catch>

<form id="welcome">
 <block>
   <prompt>Welcome to the service center of Dialogic.</prompt>
   <goto next="#directory"/>
 </block>
</form>
```

Signal after Session is Established

CPA detects any registered signal similar to CNG. It is based on the special grammar mode, "signal", and is able to detect reception of registered signals. A signal grammar built-in is available for those who need assistance using signal grammar:

```
<grammar mode="signal" src="builtin:signal/tone?SIGNAL_NAME"/>
```

On detection of a valid signal, the active field that contains the signal grammar will go into its <filled> section. The variable, application.lastresult$ or <field_name>$, will nest shadow members such as the .inputmode, the .utterance, and the .interpretation for any grammar recognition. The utterance will contain the detected signal name and the inputmode will contain the mode "signal". In the built-in grammar, the .interpretation member is the same as the .utterance member. If the grammar is fully defined with rules and tags, it will return the appropriate interpretation.

If several signals are detected, separate each signal with a ";" in the built-in signal grammar. In this instance, you could also create a built-in signal grammar for each individual signal. If a more complex signal grammar is required, it can be written as any dtmf grammar, but with the attribute mode="signal".

Example: Signal Detection using a Link Grammar with Event

```
<form id="f1">
 <field name="field1">|
   <link event="com.dlgc.signal" messageexpr="application.lastresult$.utterance">
     <grammar mode="signal" src="builtin:signal/tone?CNG"/>
   </link>
   <noinput>
     <!-- sets the value of this field, so that the FIA doesn't select this again -->
     <assign name="field1" expr="true"/>
   </noinput>
   <catch event="com.dlgc.signal ">
     <if cond="_message == 'CNG'">
     <!-- do stuff when CNG is detected -->
     </if>
   </catch>
   <!-- This field waits for the CNG tone -->
.....
 </field>
 <field name="Voicemail"/>
   <!-- When this field is visited, CNG is not enabled (the grammar is not active) -->
   <prompt> Welcome to VoiceMail. To leave a message ... </prompt>
....
 </field>
</form>
```

Example: Signal Detection Mixed with a DTMF/Voice Grammar

```
<form id="f1">
 <field name="field1" type="digits?length=4">
   <grammar mode="signal" src="builtin:signal/tone?CNG"/>
   <prompt>please enter the extension you want to reach.</prompt>
   <noinput>
     <!-- sets the value of this field, so that the FIA doesn't select this again -->
     <assign name="field1" expr="true"/>
   </noinput>
   <filled>
     <if cond="field1$.inputmode == 'signal'">
       <log>input mode is <value expr="field1$.inputmode"/> </log>
       <if cond="field1$.utterance == 'CNG'">
         <log>detected signal is <value expr="field1$.interpretation"/> </log>
         <!-- do stuff when CNG is detected -->
       <else/>
         <!-- do stuff when others signals are detected-->
       </if>
     <else/>
       <!-- do stuff for matched digits from dtmf grammar -->
       <log>input mode is <value expr="field1$.inputmode"/> </log>
     </if>
   </filled>
 </field>
</form>
```

Outbound Calling

This feature provides the capability to initiate VXML outbound calls by issuing HTTP GET requests to the internal VXML web server. Two modes of operation are provided. The flat key/value mode is the simplest to use and covers most cases. The xml mode is provided for cases that require issuing multiple calls at once. Either mode can have its responses optionally URL-encoded.

Configuration

The internal web server address, the port, and the "encode responses" parameter can be configured using the WebGUI. The firewall should be configured to accept TCP port 9002 for incoming HTTP requests.

Request URI Format

The URI used by an application server for the outbound call request is a standard HTTP GET URI format:

```
http://<xmserver_IP>:9002/callplacer?<query>
```

Set <xmserver_IP> and <query> as follows:

- <xmserver_IP> is the server IP address set on the VXML configuration page of the WebGUI.

- <query> is the request in standard URL-encoded key/value format (key1=value1&key2=value2…). The URL encoding applies to the value field only and is described in RFC 3986. If the query contains the cpxml key, the request will be in XML mode. If the query does not contain the cpxml key, the query is in flat mode.

Flat Mode

The flat request mode is a standard key1=value1&key2=value2... format. The initial request contains several keys with their encoded values that describe a single call. When using flat mode, the URI must not contain a cpxml key. Refer to the following table for keys and values.

Key	Value	Description
id	Alphanumeric string	Application supplied ID that may be used to associate the response with the call.
desturi	Destination URI	Contains the destination URI in standard SIP URI format: sip:user@destIP.
vxmluri	Initial VXML document URI	Contains the VXML initial document to execute once the destination has answered the call.
callbackuri	Application server URI	Contains a standard HTTP URL where VXML will POST the callback info.
localuser	Local user ID	Contains the user part of the FROM header sent in the outbound INVITE message.
sessionid	Session-ID SIP header	If present, sets the SIP Session-ID header. If not present, the SIP Session-ID header is automatically provided by PowerMedia XMS. The format of the value must follow RFC 7329.

Flat Mode Request Example

```
http://xmserver IP:9002/callplacer?id=call123&desturi=sip%3Auser%40remote&vxmluri=http%3A%2F%2F12
7.0.0.1%2Fvxml%2Findex.vxml&callbackuri=http%3A%2F%2Fappserver%2Fcgi%2Fvxml-
dialout.cgi&localuser=mylocalname
```

Refer to the following keys/values in clear text:

```
id=call123
```

```
desturi=sip:user@remote
```

```
vxmluri=http://127.0.0.1/vxml/index.vxml
```

```
callbackuri=http://appserver/cgi/vxml-dialout.cgi
```

```
localuser=mylocalname
```

XML Mode

XML mode can be used to place multiple calls with one request. The initial request must contain one cpxml key with the XML document as an encoded value.

Key	Value	Description
cpxml	XML document	Contains the XML document describing the outbound call requests. The XML mode may be used to specify multiple outbound call requests in the same XML document by adding several <dial> elements, each with a unique ID. This ID will be returned in the response. • <?xml version="1.0"?> • <placecall> o <dial id="X"> ▪ <desturi> ▪ <vxmluri> ▪ <callbackuri> ▪ <localuser>

Refer to the following table for parameter details.

Parameter	Value	Description
id	Alphanumeric string	Application supplied id that may be used to associate the response with the call.
desturi	Destination URI	Contains the destination URI in standard SIP URI format: sip:user@destIP.
vxmluri	Initial VXML document URI	Contains the VXML initial document to execute once the destination has answered the call.
callbackuri	Application server URI	Contains a standard HTTP URL where VXML will POST the callback info.
localuser	Local user ID	Contains the user part of the FROM header sent in the outbound INVITE message.
sessionid	Session-ID SIP header	If present, sets the SIP Session-ID header. If not present, the SIP Session-ID header is automatically provided by PowerMedia XMS. The format of the value must follow RFC 7329.

XML Mode Request Example

```
http://xmserver_IP:9002/callplacer?cpxml=url-encoded-xml-document
```

Refer to *url-encoded-xml-document* in clear text:

```
<?xml version="1.0"?>
<placecall>
    <dial id="1">
        <desturi>sip:user@remote</desturi>
        <vxmluri>http://127.0.0.1:9002/vxml/index.vxml</vxmluri>
        <callbackuri>http://appserver/cgi/vxml-dialout.cgi</callbackuri>
        <localuser>mylocalname</localuser>
    </dial>
</placecall>
```

Response and Callback

A response is an immediate 200 OK acknowledgement to an initial HTTP GET request. A callback is an asynchronous HTTP POST sent to the callbackuri URL specified in the request with call progress information.

Note: For parsing, do not assume the order and the number of the keys in the response and callback messages.

Returned Fields

Responses and callbacks return some fields that indicate the call state.

Field	Presence	Description
id	Always	Application supplied id from the call request that may be used by the application to associate the response with the call.
type	Always	The reason of the message. Refer to the Types table that follows.
callid	If no error	The PowerMedia XMS call id of the outbound call.
sessionid	If no error	The SIP Session-ID returned from the original request or provided automatically by PowerMedia XMS.
status	Only with type CALLENDED	Cause of the call termination. Currently always OK.

Types

Response and callback can return different type.

Field	Presence	Description
response	DIALING	The call is initiated.
	NORESOURCE	No resource to place the call.
	UNAUTHORIZED	The call not authorized.
	ERROR	An error occurred. See the logs for details.
callback	ANSWERED	Remote answered the call.
	IN-CALL	The call is active (sent every 15s).
	CALLENDED	The call has ended.
	ERROR	An error occurred. See the logs for details.

Encoding and Content-Type Configuration

The response and callback contents are sent with a Content-Type depending on the requested mode (XML or flat) and on the encoding selected with the "encode responses" parameter on the VXML page in the WebGUI.

Select the "encode responses" parameter on the VXML page in the WebGUI to enable standard URL-encoding (%XX) as described in RFC 3986 (e.g., key=value&key=value… with the value URL-encoded). Deselect the **encode responses** parameter to disable URL-encoding. URL-encoding is disabled by default.

Refer to the following table for details.

Encoding	Mode	Content-Type	Body Format
Enabled	XML	application/x-www-form-urlencoded	cpxml=*encoded-xml* *encoded-xml* is an xml document URL-encoded with the %XX (RFC 3986). Refer to the XML Elements table.
	Flat	application/x-www-form-urlencoded	type=*encoded-type*&callid=*encoded-callid*&status=OK *encoded-type* and *encoded-callid* are values URL-encoded with the %XX (RFC 3986).
Disabled	XML	text/xml	Text xml document. Refer to the XML Elements table.
	Flat	text/plain	Line based format: - type=value\r\n - id=value\r\n - callid=value\r\n - sessionid=value\r\n - status=value\r\n

XML Elements

Return	Description

Response	Contains as many XML <response> elements as requested <dial> elements, each of them associated to a unique ID. Once the call ID is obtained, it will be used as reference id for the call, and the ID attribute will no longer be used.
	
	• <?xml version="1.0"?>
	• <placecall>
	   o <response type=T id=X> *(X is the ID of the corresponding <dial> request)*
	      ▪ <callid>
	      ▪ <sessionid>
	      ▪ <status>

Callback	Contains the XML callback.
	
	• <?xml version="1.0"?>
	• <placecall>
	   o <callback type=T id=X>
	      ▪ <callid>
	      ▪ <sessionid>
	      ▪ <status>

Example of URL-Encoded XML Content in 200 OK Response

```
cpxml=%3c%3fxml%20version%3d%221.0%22%3f%3e%3cplacecall%3e%3cresponse%20type%3d%22DIALING%22%20id
%3d%221%22%3e%3ccallid%e528d90a-a271-4aeb-8f19-
789d784c863c%3c%2fcallid%3e%3csessionid%1f438da63879438387c4042f97bd8a19%3c%2fsessionid%3e
%3c%2fresponse%3e%3c%2fplacecall%3e
```

Example of Text/XML Content in 200 OK Response

```xml
<?xml version="1.0"?>
<placecall>
    <response type="DIALING" id="1">
        <callid>e528d90a-a271-4aeb-8f19-789d784c863c</callid>
        <sessionid>1f438da63879438387c4042f97bd8a19</sessionid>
    </response>
</placecall>
```

Example of Text/XML Content in POST Callback

```xml
<?xml version="1.0"?>
<placecall>
    <callback type="ANSWERED" id="1">
        <callid>e528d90a-a271-4aeb-8f19-789d784c863c</callid>
        <sessionid>1f438da63879438387c4042f97bd8a19</sessionid>
    </callback>
</placecall>
```

Example of Text/Plain Flat Content in Response or Callback

```
type=DIALING\r\n
id=1\r\n
callid=5a228da1-4cee-47eb-a5e9-942e5582ab42\r\n
sessionid=1f438da63879438387c4042f97bd8a19\r\n
```

Example of URL-Encoded Flat Content in Response or Callback

```
type=ANSWERED&id=1&callid=5a228da1-4cee-47eb-a5e9-942e5582ab42&
sessionid=1f438da63879438387c4042f97bd8a19
```

Behavior Changes

The following behavior changes have been implemented in VoiceXML.

Note: The behavior changes affect PowerMedia XMS Release 2.2 Service Update 1 and later.

- When a VoiceXML script exits or disconnects with <exit> reason, the BYE content will encode special characters.

  For instance, exit a VoiceXML script with the following:
  ```
  <exit expr="'post_process_exit_expr'"/>
  ```

  XMS 2.2 SU1 and later: _exit=post_process_exit_expr&_reason=exit
  XMS 2.2 and earlier: _exit=post%5Fprocess%5Fexit%5Fexpr&_reason=exit

- When sending re-INVITE hold during VoiceXML script processing, it will pause during the media execution and continue once resume is sent.

  XMS 2.2 SU1 and later: Upon sending re-INVITE hold, it will pause during the media execution and continue once resume is sent.
  XMS 2.2 and earlier: Upon sending re-INVITE hold, it will finish the media execution then pause before the next media followed.

Restrictions and Limitations

The following restrictions and limitations may affect VoiceXML.

- When a call leg issuing a <transfer> is hung up in VoiceXML, it cannot have any application supplied data in the BYE. For example, if an <exit> expr is supplied in a <catch> event for "connection.disconnect.transfer", the expr or namelist supplied will not be included in the BYE.
  ```
  <catch event="connection.disconnect.transfer">
         <exit expr="Call transfer completed"/>
  </catch>
  <form>
         <transfer bridge="false" dest="xxx"/>
  </form>
  ```

4.   Alphabetical VoiceXML Reference

Note: When the default value is "N/A," there is no static default value because the default value might depend on context (e.g., the default value might can change based on the higher level property).

<assign>

Assigns a value to a variable.

Syntax

```
<assign
   expr = "PCDATA"
   name = "identifier"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
expr	PCDATA	yes	N/A	ECMAScript expression to evaluate and assign to the named variable.
name	identifier	yes	N/A	Name of the variable. This must be a legal ECMAScript identifier. By default, the variable is resolved within the closest enclosing scope of the active element. To remove ambiguity, prefix the variable name with a scope name.

Details

You must explicitly declare a variable using a <var> element or <var> statement within a <script> element. Attempting to assign a value to an undeclared variable causes the VoiceXML Interpreter to return an error.semantic error.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
<var name="number"/>
<var name="month" expr="'March'"/>
    <form>
        <block>
            Month is <value expr="month"/>
            <assign name="month" expr="'July'"/>
            Now month is <value expr="month"/>
```

```
            <assign name="number" expr="2*5"/>
            Number is <value expr="number"/>
        </block>
    </form>
</vxml>
```

See Also

<script>, <var>

<audio>

Plays the media file at the specified location.

Syntax

```
<audio
    src = "URI"
    expr = "CDATA"
    fetchhint = "{prefetch | safe}"
    fetchtimeout = "CDATA"
    maxage = "CDATA"
    maxstale = "CDATA"
    offset = "CDATA"
    offsetexpr = "CDATA"
    dlgc:type = "CDATA"
    dlgc:playcontrol = "boolean"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
dlgc:playcontrol	boolean	no	false	Enable or disable the playback control for that audio element. Valid values: • true • false Attribute is optional. If not present, playback control is disabled for the audio element.
expr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI of the media file. Programmable Media Platform proprietary.

Attribute	Data Type	Required?	Default	Description
fetchhint	{prefetch | safe}	no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: • prefetch = Fetch the resource when the page is loaded. • safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh. If no maxage or maxstale values are set, the VoiceXML Interpreter: • Uses the cached resource, if the resource remains in the cache. • Performs a normal fetch, if the resource is not in the cache.
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.

Attribute	Data Type	Required?	Default	Description
offset	CDATA	no	N/A	Programmable Media Platform proprietary. The offset (in s or ms) at which to play the media file.
offsetexpr	CDATA	no	N/A	An ECMAScript expression that evaluates to the offset. This can be used in conjunction with the lastresult$.bargeintime variable to let the application play a media file from the point at which the user barged in, and not have to restart the media file. This is especially useful for long media files.
src	URI	no	N/A	URI of the media file.
dlgc:type	CDATA	no	N/A	Optionally specifies the media type of the requested resource when the resource type cannot be derived from a VXML recording or the specified resource URI file extension. When a recording file is specified in the attribute of a <record> element, it is not recommended to specify the dlgc:type attribute because the type will be automatically taken from the recording data. If the dlgc:type attribute is specified, the specified dlgc:type attribute will take precedence. When using the phrase server that returns a list of URI, the type should be text/URI-list. Valid values: • audio/x-wav = WAV (RIFF header). • audio/x-aud = Dialogic proprietary.

Attribute	Data Type	Required?	Default	Description
				• audio/basic = Raw (headerless). • audio/x-alaw-basic = Raw (headerless). • audio/L8 = Valid rates are 8000, 11025, and 16000 (e.g., audio/L8; rate=16000). • audio/L16 = Valid rates are 8000, 11025, and 16000 (e.g., audio/L16; rate=16000). • audio/amr = Adaptive Multi-Rate (amr). • audio/amr-wb = Adaptive Multi-Rate Wideband (G.722.2/ amr-wb). • audio/3gpp = AMR or AMR-WB. The mode=xxx, codec=xxx, and rate=yyy specifications are not required on playback for file types that have headers such as .wav, .aud, and .3gp.

Details

The <audio> element can contain child content (alternate content) that is played if the media file specified by the src or expr attributes is either not found or cannot be played. The <audio> element can reference an HTTP, file, or builtin in its src attribute. For example:

- file://vxml/greeting.wav
- builtin:audio/nomatch

Valid source file types are:

- Raw/headerless (.alaw and .ulaw)
- RIFF (.wav)

VoiceXML is capable of generating DTMF (RFC 2833) with the <audio> element. To generate DTMF, use the following example syntax:

```
<audio src="builtin:audio/dtmf2833/dtmf-string"/>
```

In the above example, replace "dtmf-string" with a string of DTMF digits from the set: 0123456789*#ABCD

Parents

<audio>, <block>, <catch>, <choice>, <enumerate>, <error>, <field>, <filled>, <foreach>, <help>, <if>, <initial>, <media>, <menu>, <noinput>, <nomatch>, <object>, <prompt>, <record>, <subdialog>, <transfer>

Children

<audio>, <break>, <desc>, <emphasis>, <enumerate>, <mark>, <media>, <p>, <phoneme>, <prompt>, <prosody>, <s>, <say-as>, <sub>, <value>, <voice>

Example

The following example plays a .wav file:

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <prompt>
                <audio src="../audio/test.wav" maxage="0"/>
            </prompt>
            <prompt>
                <audio src="http://www.webserver.com/doesNotExist.wav"
                    fetchtimeout="1s">
                    Audio does not exist
                </audio>
            </prompt>
        </block>
    </form>
</vxml>
```

See Also

<prompt>

<block>

A form item specifying a sequence of items to execute (executable content).

Syntax

```
<block
   cond = "CDATA"
   expr = "CDATA"
   name = "identifier"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	no	ECMAScript undefined	Boolean expression that must evaluate to ECMAScript true for the block to execute.
expr	CDATA	no	N/A	Initial value of the named form item variable. If initialized to a value, then the form item is not visited unless the form item variable is cleared.
name	identifier	no	N/A	Name of the form item variable. This must be a legal ECMAScript identifier.

Details

The contained items are executed if:

- form item variable of the block is undefined, and
- cond attribute, if specified, evaluates to true

The form item variable is automatically set to true just before the block is entered.

Parents

<form>

Children

<assign>, <audio>, <clear>, <data>, <disconnect>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <var name="bCond" expr="false"/>
    <form id="form1">
        <block name="block1" cond="bCond">
            <prompt>Currently in block one</prompt>
        </block>
        <block name="block2">
            <prompt>Now in block two</prompt>
        </block>
    </form>
</vxml>
```

See Also

<form>

<break>

Inserts a pause or prosodic boundary between words.

Syntax

```
<break
    strength = "{x-weak | weak | medium | strong | x-strong | none}"
    time = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
strength	identifier	no	medium	Strength of the prosodic break in the speech output. Valid values: • medium • none • strong • x-strong • x-weak • weak
time	CDATA	no	N/A	Duration to pause. Specify s for seconds and ms for milliseconds.

Details

If a <break> element is not used with strength or time attributes, the VoiceXML Interpreter produces a break with a prosodic strength greater than that which the processor would otherwise have used if no break element was supplied.

Parents

<audio>, <choice>, <emphasis>, <enumerate>, <option>, <p>, <prompt>, <prosody>, <s>, <voice>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <prompt>
                This is a default medium pause:
                <break/>
                This is a ten millisecond pause:
                <break time="10ms"/>
                This is an extra-strong break:
                <break strength="x-strong"/>
            </prompt>
        </block>
    </form>
</vxml>
```

<audio>, <prompt>

<catch>

Defines an event handler to catch events raised by the application.

Syntax

```
<catch
    event = "identifiers"
    cond= "CDATA"
    count = "integer"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
event	identifiers	yes	N/A	Event to handle. Multiple events can be caught in a single catch statement using a space-separated namelist.
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the catch to execute.
count	integer	yes	N/A	Numerical occurrence of the catch event, such as 2 for the second occurrence. This allows you to handle different occurrences of the catch event in different ways.

Details

A catch contains executable content to run when a specific event is raised. Use the following shorthand elements catch common events:

Element	Equivalent <catch> Syntax
<error>	<catch event="error">
<help>	<catch event="help">
<noinput>	<catch event="noinput">
<nomatch>	<catch event="nomatch">

The <catch> element includes two variables in anonymous scope:

- _event contains the name of the event that was raised.
- _message contains the message accompanying the raised event.

Parents

<field>, <form>, <initial>, <menu>, <object>, <record>, <subdialog>, <transfer>, <vxml>

Children

<assign>, <audio>, <clear>, <data>, <disconnect>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <reprompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```xml
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <catch event="terminate">
        Caught terminate
        <goto next="#formExit"/>
    </catch>
    <error>
        Caught error
        <!-- Log the exact error that was caught: -->
        <log label="WARN">
            The exact error was: <value expr=" event"/>
        </log>
        <goto next="#formExit"/>
    </error>
    <var name="bCondition" expr="false"/>
    <form id="formEntry">
        <!-- A handler for two kinds of event: -->
        <catch event="event.foo event.bar">
            Form scope event handler
            <log label="INFO">
                The event thrown was: <value expr=" event"/>.
                The message is: <value expr=" message"/>.
            </log>
            Throwing terminate
            <throw event="terminate"/>
        </catch>

        <catch event="terminate" cond="bCondition">
            This handler will not be executed due to the condition
        </catch>
        <block>
            Throwing custom event
            <throw event="event.foo" message="Thrown from formEntry"/>
        </block>
    </form>
    <form id="formExit">
        <block>
            Goodbye
            <exit/>
        </block>
    </form>
</vxml>
```

See Also

<help>, <error>, <nomatch>, <noinput>

<choice>

Defines a choice item in a menu.

Syntax

```
<choice
   accept = "{exact|approximate}"
   dtmf = "CDATA"
   event = "identifier"
   eventexpr = "CDATA"
   expr = "CDATA"
   fetchaudio "URI"
   fetchhint = {prefetch | safe}
   fetchtimeout = "CDATA"
   maxage "CDATA"
   maxstale "CDATA"
   message "CDATA"
   messageexpr "CDATA"
   next"URI"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
accept	{exact | approximate}	no	exact	Determines how the choice is activated. Valid values: • approximate = Utterances containing a sub-phrase of the choice phrase activate the choice. • exact = Utterances that match the entire choice phrase activate the choice.
dtmf	CDATA	no	N/A	DTMF grammar assigned to the choice.
event	identifier	no	N/A	Event raised by the choice.
eventexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the event to raise.
expr	CDATA	no	N/A	ECMAScript expression that evaluates to the next URI to load.

Attribute	Data Type	Required?	Default	Description
fetchaudio	URI	no	N/A	URI of an audio resource to play while the XML data is fetched. If the fetchaudio attribute is not set, no audio is played during the fetch. If the audio is playing after the XML document is fetched, the audio terminates.
fetchhint	{prefetch | safe}	no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: • prefetch = Fetch the resource when the page is loaded. • safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh.
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.
message	CDATA	no	N/A	Additional information about the event being raised. The message is available as the ECMAScript variable _message in the <catch> element that handles the event.

Attribute	Data Type	Required?	Default	Description
messageexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the message.
next	URI	no	N/A	URI of the next dialog or document to load.

Details

You can associate speech, DTMF grammars, or both with the choice. When the choice is selected, the VoiceXML Interpreter can:

- Jump to the URI specified by next (or the URI obtained from the ECMAScript expression expr).

- Raise an event (specified either by event, or by evaluating the ECMAScript expression eventexpr).

Parent

<menu>

Children

<audio>, <break>, <enumerate>, <grammar>, <mark>, <p>, <phoneme>, <prosody>, <s>, <say-as>, <value>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <catch event="quitEvent">
        Goodbye
        <exit/>
    </catch>
    <menu id="main">
        <prompt>What do you want to eat, apple or orange?</prompt>
            <choice dtmf="1" next="#formApple"> apple </choice>
            <choice expr="'#' + 'formOrange'"> orange </choice>
            <choice event="quitEvent"> quit </choice>
            <nomatch>Please say apple, orange, or quit</nomatch>
    </menu>
        <form id="formApple">
            <block>
                You chose apple
                <exit/>
            </block>
        </form>

        <form id="formOrange">
            <block>
                You chose orange
                <exit/>
            </block>
        </form>
</vxml>
```

See Also

<enumerate>, <link> , <menu>, <option>

<clear>

Resets one or more variables, including form items.

Syntax

```
<clear
    namelist = "form_item_name_1 form_item_name_2 form_item_name_3 ..."
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
namelist	list of identifiers	no	medium	Space-separated list of form item variables. If not specified, all form item variables in the current form are cleared.

Details

Resetting a variable does both of the following:

- Sets to undefined the variables specified by namelist.
- Re-initializes the prompt and event counters.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <field name="pin" type="digits?length=4">
            <prompt>What is your pin number?</prompt>
            <noinput>
                Please say or key in your four digit pin number
            <reprompt/>
            </noinput>
            <filled>
                <if cond="pin == 1234">
                    Code accepted
                <else/>
                    Invalid pin - please try again
                    <clear namelist="pin"/>
                </if>
            </filled>
        </field>
    </form>
</vxml>
```

See Also

<form>

<data>

Enables a VoiceXML application to fetch arbitrary XML data from and submit data to, a document server, without transitioning to a new VoiceXML document. The XML data fetched by the <data> element is bound to ECMAScript through the named variable that exposes a read-only subset of the XML DOM.

Syntax

```
<data
    enctype = "CDATA"
    fetchaudio = "URI"
    fetchhint = "{prefetch | safe}"
    fetchtimeout = "CDATA"
    maxage = "CDATA"
    maxstale = "CDATA"
    method = "{get | post}"
    name = "identifier"
    namelist = "list of identifiers"
    src = "URI"
    srcexpr = "CDATA"
/>
```

Attributes

None of the attributes for <data> are required.

Attribute	Data Type	Required?	Default	Description
enctype	CDATA	no	application /x-www- form- urlencoded	MIME encoding type to use when submitting data. Specify a MIME type of multipart/form-data to upload the audio file that results from a record operation.
fetchaudio	URI	no	N/A	URI of an audio resource to play while the XML data is fetched. If the fetchaudio attribute is not set, no audio is played during the fetch. If the audio is playing after the XML document is fetched, the audio terminates.

Attribute	Data Type	Required?	Default	Description
fetchhint	{prefetch | safe}	no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: • prefetch = Fetch the resource when the page is loaded. • safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh.
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.
method	CDATA	no	get	HTTP request method. Valid values: • get = Execute HTTP get. • post = Execute HTTP post.
name	identifier	no	N/A	Name of the variable that exposes the XML DOM. This must be a legal ECMAScript identifier.

Attribute	Data Type	Required?	Default	Description
namelist	list of identifiers	no	N/A	Space-separated list of variables submitted. By default, no variables are submitted.
src	URI	no	N/A	URI that specifies the location of the XML data to retrieve.
srcexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI specifying the location of the XML data to retrieve.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <form>, <help>, <if>, <noinput>, <nomatch>, <vxml>

Children

None.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
    <quote>
        <ticker>F</ticker>
        <name>Ford Motor Company</name>
        <change>1.00</change>
        <last>30.00</last>
     </quote>
VoiceXML that retrieves a stock quote from the XML document:

<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">

    <data name="quote" src="quote.xml" maxage="0"/>
    <script>
        var price = quote.documentElement.getElementsByTagName("last").
            item(0).firstChild.data;
    </script>
    <form>
        <block>
            Price is <value expr="price"/>
        </block>
    </form>
</vxml>
```

See Also

<script>, <submit>, <var>

<desc>

Provides a textual description of the referenced audio source.

Syntax

```
<desc
   xml:lang = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
xml:lang	CDATA	no	N/A	RFC 1766 compliant identifier used to indicate that the content of the element is in a different language from that of the content surrounding the element.

Details

This element can only occur as a child of <audio>. It does not affect the audio output in any way.

Parents

<audio>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <audio src="../audio/effect_1.wav">
                <desc> Vivaldi music earcon </desc>
            </audio>
        </block>
    </form>
</vxml>
```

<disconnect>

Causes the VoiceXML Interpreter context to disconnect from the user.

Syntax

```
<disconnect
   namelist = "list of identifiers"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
namelist	list of identifiers	no	N/A	Space-separated list of form item variable to return to the VoiceXML Interpreter context. By default, no variables are returned.

Details

Causes a connection.disconnect.hangup event to occur.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <catch event="connection.disconnect.hangup">
        <log expr="'disconnected'"/>
        <exit/>
    </catch>
    <form>
        <block>
            <prompt>Get ready to disconnect!</prompt>
            <disconnect/>
        </block>
    </form>
</vxml>
```

See Also

<exit>

<dlgc:file>

The file operation to be executed on the specified file(s): copy, move, delete, or append.

The following declaration must be used to enable the dlgc domain name:

```
xmlns:dlgc="http://www.dialogic.com/xmlns"
```

Syntax

```
<dlgc:file
   name = "identifier"
   op = "CDATA"
   src = "URI"
   srcexpr = "CDATA"
   dest = "URI"
   destexpr = "CDATA"
   xml:lang = "CDATA"
   overwrite = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
name	identifier	yes	N/A	File variable name.
op	CDATA	yes	N/A	The operation to perform on the file(s). One of the following: copy, move, delete, or append.
src	URI	yes (exclusive with srcexpr)	N/A	URI of the source file.
srcexpr	CDATA	yes (exclusive with src)	N/A	ECMAScript expression that evaluates to the URI of the source file.
dest	URI	no	N/A	URI of the transfer destination.
destexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI of the destination file.
xml:lang	CDATA	no	N/A	Language for the media relative path.
overwrite	CDATA	no	yes	If set to no, the file operation will not overwrite the destination file if it exists. • yes (default) • no

Details

If present, the xml:lang should take precedence over any other record locale declaration at the document or configuration level.

The src attribute is a URI and can be relative path file://, absolute path file:///, external http://, or phrasal server var://.

After the operation is completed, the name of the element should reference the src or dest file according the operation.

If the operation is successful, shadow variables are created:

- myname$.success is set to "true".
- myname$.status should contain the reason ("0 No Error").
- myname$.http should contain the http URI to the file. This shadow variable should be undefined if the URI is an absolute file path.

If the operation fails, the shadow variables myname$.success is set to "false" and myname$.status should contain the reason. Other shadow variables have an undetermined value. No error is thrown in the VoiceXML Interpreter.

The failure is logged in the app log file (and in the general VoiceXML log).

This variable has the same characteristics as a record variable and can be for instance, played or submitted.

Operations

copy

This operation creates a copy specified in the dest attribute from the file specified in the src attribute. After the operation, the variable name should reference the dest URI.

Example:
```
<dlgc:file name="myfile" xml:lang="en-US" src="file://myfile.wav" dest="file://save/mycopy.wav"
op="copy" />
```

Copy the myfile.wav to saved/mycopy.wav. It uses the local xml:lang = en-US to construct the relative media path.

Submit Content Usage:
```
<submit next="nexturi" namelist="myfile" />
```

Submit HTTP URI Only:
```
<var name="myhttp" expr="myfile$.http" />
<submit next="nexturi" namelist="myhttp" />
```

move

This operation moves the file specified in the src attribute to the file specified in the dest attribute. After the operation, the variable name should reference the dest URI.

Example:
```
<dlgc:file name="myfile" src="file://myfile.wav" op="move" />
```
Move the myfile.wav to saved/mymove.wav.
```
<dlgc:file name="myfile" src="file://myfile.wav" op="move" />
```
Move the myfile.wav to saved/mymove.wav.

delete

This operation deletes the file specified in the src attribute. After the operation, the variable name should reference the dest URI and the shadow variables size and duration should not contain significant information.

Example:
```
<dlgc:file name="myfile" xml:lang="en-US" src="file://myfile.wav" op="delete" />
```
Delete the myfile.wav.

append

This operation appends the files listed in the src attribute to the single file specified in the dest attribute. After the operation, the variable name should reference the dest URI.

Multiple src files must be separated by a new line character "\n". If the dest file does not exist, it is created.

Example:

```
<dlgc:file name="myfile" xml:lang="en-US" src="file://header/myheader.wav\nfile://myfile.wav"
dest=file://myfilewithheader.wav op="append" />
```

Append the file://header/myheader.wav and file://myfile myfile.wav to file://myfile.wav.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

None.

See Also

<record>, <submit>

<dlgc:recordcall>

As part of the multitrack record feature, the <dlgc:recordcall> element handles the recording of call streams for transmit and receive directions in separate tracks of a container referenced in the destination URI. It is an executable element that works asynchronously (does not wait for the recording to finish in order to complete the element execution).

Note: The <dlgc:recordcall> element utilizes the multitrack record feature of the underlying media engine subsystem (HMP). While VXML itself does not limit the specification of recording formats, support is defined by the underlying media engine subsystem formats supported by the multitrack record feature. PowerMedia XMS currently supports only WAV container with two audio tracks.

The following script must be used to declare the dlgc namespace attribute in the <vxml> element:

```
xmlns:dlgc=http://www.dialogic.com/xmlns
```

Syntax

```
<dlgc:recordcall
   name = "identifier"
   op = "identifier"
   direction = "identifier"
   dest = "CDATA"
   destexpr = "CDATA"
   type = "CDATA"
   videotype = "CDATA"
   xml:lang = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
name	identifier	yes	N/A	The input item variable that will hold the recording.
op	identifier	no	start	The operation to perform. Valid values: • start = Start a call recording. • stop = Stop the current call recording. • update = Update the direction attribute.
direction	identifier	no	sendrecv	Specifies the direction (stream) that will be recorded. Valid values: • sendrecv = Record both directions in separate tracks. • sendonly = Record only the send side (the prompt play/tts/destination party of bridge transfer). Record silence in the other track. • recvonly = Record only the receive side (the caller). Record silence in the other track. • inactive = Fill the tracks with silence.
dest	CDATA	no	if not supplied, a file name will be generated	The URI of the destination file. • If set, specifies the file name to save the recording. The file will not be deleted at the end of the session. • If not set, file is automatically created and will be deleted at the very end of session (after post processing).

Attribute	Data Type	Required?	Default	Description
destexpr	CDATA	no	N/A	ECMAScript for the destination file URI. If set, overrides the dest attribute.
type	CDATA	no	audio/x-wav with PowerMedia XMS default codec	Specifies the audio type. Valid values: • audio/x-wav • audio/3gpp Currently not supported. • audio/mp4 Currently not supported. • audio/x-matroska Currently not supported. **Note:** The media engine subsystem may not support all of the listed audio types for multitrack recording.
videotype	CDATA	no	N/A	Specifies the video type. Valid values: • video/3gpp Currently not supported. • video/mp4 Currently not supported. • video/x-matroska Currently not supported. **Note:** The media engine subsystem may not support all of the listed video types for multitrack recording.
xml:lang	CDATA	no	N/A	Specifies the language and locale of the document. **Note:** xml:lang is ignored if using an absolute path.

Details

The element starts a call recording that will end either by stopping it manually using the op=stop attribute or automatically on hangup (remote caller or local <disconnect>). The recording can be handled like a standard <record> using the name attribute. It can be played or submitted.

A standard VXML script can be run during call recording, including standard VXML <record>, <audio>, or <field> form items.

If there is no dlgc:dest or dlgc:destexpr destination attribute specified, the system will create a temporary recording file and delete it when the channel is released.

At the end of recording, call recording information will be available in name$.duration and name$.size. In addition, name$.http will contain the HTTP URI of the recorded file if it was generated automatically (dest and destexpr left unset).

Only one multi-track call recording can be active on a given call.

Audio/Video Attribute Types

audio/video	not set	empty	filled
not set	audio/x-wav, no video	N/A	audio/*
empty	audio/x-wav, no video	N/A	audio/*
filled	video/*	video/*	audio/*, video/* (same sub-name)

N/A = Not Allowed

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml xmlns="http://www.w3.org/2001/vxml" xmlns:dlgc="http://www.dialogic.com/xmlns"
version="2.1">
<form>
        <catch event="connection.disconnect.hangup">
                <goto next="#end"/>
        </catch>

        <field name="entry" type="digits?length=4">
                <prompt>Welcome to ACME company. Please say or enter the extension you wish to
reach.</prompt>
                <filled>
                        <prompt>The conversation will be recorded to improve the quality of the
customers service.</prompt>
                        <dlgc:recordcall name="myrecordcall">
                </filled>
        </field>
        <transfer name="mytransfer" type="bridge" dest="sip:agentqueue@callcenter">
                <filled>
                        <dlgc:recordcall name="myrecordcall" op="stop">
                        <goto next="#end"/>
                </filled>
```

```
        </transfer>
<form>

<form id="#end">
        <block>
                <submit next="http://mysubmitserver/vxml/submit.php" method="post"
enctype="multipart/form-data" namelist="myrecordcall"/>
                </exit>
        </block>
</form>
</vxml>
```

<else>

Used in if..elseif..else conditional logic.

Syntax

```
<if cond="CDATA">
   <!-- do something -->
<elseif cond="CDATA"/>
    <!-- do something else -->
<elseif cond="CDATA"/>
    <!-- do something else -->
<else/>
   <!-- do something else -->
</if>
```

Attributes

None.

Parents

<if>

Children

None.

Example

```
<?xml version="1.0"?>
    <vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <field name="pin" type="digits?length=4">
            <prompt>What is your pin number?</prompt>
            <noinput>
                Please say or key in your four digit pin number
                <reprompt/>
            </noinput>
            <filled>
                <if cond="pin == 1234">
                    Code accepted
                <elseif cond="pin == 4321">
                    Code accepted
                <else/>
                    Invalid pin - please try again
                    <clear namelist="pin"/>
                </if>
            </filled>
        </field>
    </form>
</vxml>
```

See Also

<elseif>

Used in if..elseif..else conditional logic.

Syntax

```
<if cond="CDATA">
   <!-- do something -->
<elseif cond="CDATA"/>
  <!-- do something else -->
<elseif cond="CDATA"/>
  <!-- do something else -->
<else/>
<!-- do something else -->
</if>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	yes	N/A	Boolean expression that must evaluate to ECMAScript true for the <elseif> element to execute.

Parents

<if>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <field name="pin" type="digits?length=4">
            <prompt>What is your pin number?</prompt>
            <noinput>
                Please say or key in your four digit pin number
                <reprompt/>
            </noinput>
            <filled>
                <if cond="pin == 1234">
                    Code accepted
                <elseif cond="pin == 4321"/>
                    Code accepted
                <else/>
                    Invalid pin - please try again
                    <clear namelist="pin"/>
                </if>
            </filled>
        </field>
    </form>
</vxml>
```

<else>, <if>

<emphasis>

Specifies the stress of the text enclosed within the <emphasis> element.

Syntax

```
<emphasis
    level="{strong|moderate|reduced|none}"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
level	{strong | moderate | reduced | none}	yes	moderate	Amount of emphasis assigned to the text. Valid values: • strong • moderate • reduced • none

Parents

<audio>, <choice>, <emphasis>, <enumerate>, <option>, <p>, <s>, <prompt>, <prosody>, <voice>

Children

<audio>, <break>, <emphasis>, <enumerate>, <mark>, <phoneme>, <prosody>, <say-as>, <value>, <voice>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            Thank you <emphasis>so</emphasis> much.
        </block>
    </form>
</vxml>
```

See Also

<prompt>

<enumerate>

Generates an automatic prompt of menu choices or field options.

Syntax

Attributes

None.

Details

The <enumerate> element can be empty or it can contain text, speech mark-up, and <value> elements that reference the following special variables:

- _prompt contains the text for the current choice or option.
- _dtmf contains the DTMF sequence for the current choice or option.

Parents

<audio>, <block>, <catch>, <choice>, <enumerate>, <error>, <field>, <filled>, <foreach>, <help>, <if>, <initial>, <menu>, <noinput>, <nomatch>, <object>, <prompt>, <prosody>, <record>, <subdialog>, <transfer>, <voice>

Children

<audio>, <break>, <emphasis>, <enumerate>, <mark>, <p>, <phoneme>, <prosody>, <say-as>, <s>, <value>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <catch event="myEvent">
        Goodbye
        <exit/>
    </catch>
    <catch event="nomatch">
        I didn't understand. Say one of <enumerate/>
    </catch>
<!--
    Note that the DTMF sequence for each choice will be
    assigned automatically
-->
    <menu id="main" dtmf="true">
        <prompt>
            Main Menu <break time="500ms"/>
            <enumerate>
            For <value expr="_prompt"/> say <value expr="_prompt"/>
                or press <value expr="_dtmf"/>
            </enumerate>
        </prompt>
        <choice next="#news"> news </choice>
        <choice next="#sport"> sport </choice>
        <choice event="myEvent"> quit </choice>
    </menu>
    <form id="news">
        <block>
            You chose news
            <goto next="#main"/>
        </block>
    </form>
    <form id="sport">
        <block>
            You chose sport
            <goto next="#main"/>
        </block>
    </form>
</vxml>
```

See Also

<choice>, <option>

<error>

Catches the error event.

Syntax

```
<error
   cond = "CDATA"
   count = "integer"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the catch to execute.
count	integer	no	N/A	Occurrence number of the error. This permits alternate handlers to be defined for the error. For example, count=5 specifies the fifth occurrence of the error.

Details

This element is shorthand for <catch event="error">.

Parents

<field>, <form>, <initial>, <menu>, <record>, <subdialog>, <transfer>, <vxml>

Children

<assign>, <audio>, <clear>, <data>, <disconnect>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <reprompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <error>
        An error has occurred -- please call again later.
        <exit/>
    </error>
    <form id="formEntry">
        <block>
            Good applications are written defensively
        </block>
    </form>
</vxml>
```

See Also

<catch>

<example>

Defines an example phrase within XML grammar constructs.

Syntax

Attributes

None.

Details

The <example> element is part of the Speech Recognition Grammar Specification (SRGS). Use the example phrase to create a successful recognition result for the grammar in question.

Parents

<rule>

Children

None.

Example

```
<?xml version= "1.0"?>
<grammar xmlns="http://www.w3.org/2001/06/grammar"
    xml:lang="en-GB" root="rule1">
    <rule id="rule1" scope="public">
        <example> irish </example>
        <example> or, english </example>
        <one-of>
            <item> irish
                <tag><fieldone "irish"></tag>
            </item>
            <item> english
                <tag><fieldone "english"></tag>
            </item>
        </one-of>
    </rule>
</grammar>
```

<exit>

Terminates all loaded documents in the application.

Syntax

```
<exit
    expr = "CDATA"
    namelist = "list of identifiers"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
expr	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the exit to execute.

Attribute	Data Type	Required?	Default	Description
namelist	list of identifiers	no	N/A	Space-separated list of form item variables to return. If not specified, no variables are returned.

Details

When <exit> is executed, control is returned to the VoiceXML Interpreter. An exit event is not raised.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="formExit">
        <block>
            <prompt>Thanks for using the voice portal</prompt>
            <exit/>
        </block>
    </form>
</vxml>
```

See Also

<disconnect>

<field>

Specifies an input field for a form.

Syntax

```
<field
   cond = "CDATA"
   expr = "CDATA"
   modal = "{true|false}"
   name = "identifier"
   slot = "identifier"
   type = "{boolean |  currency |date |digits | number | phone | time}"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the <field> element to execute.
expr	CDATA	no	ECMAScript undefined	Initial value of the form item variable.
modal	boolean	no	false	Grammars to enable. Valid values: • true = Only the field's grammars are enabled. All other active grammars are temporarily disabled. • false = All active grammars are enabled.
name	identifier	no	N/A	Name of a shadow variable or a field item variable that holds the recognition result. Shadow variables are values available from a field item, specified as fieldItemVariableName$.shadow Variable. Valid values: • name$.bargeintime **=** Time interval between playback started and bargein occurred, in ms (Programmable Media Platform proprietary). • name$.confidence = Confidence level in the recognized result. Valid values range from 0.0 (minimum confidence level) to 1.0 (maximum confidence level). • name$.interpretation **=** ECMAScript variable containing the interpretation of the utterance. • name$.inputmode = Mode in which user input was provided. Valid values are either dtmf or voice. • name$.recording = Reference to the utterance recorded during recognition, if the input mode was voice; otherwise undefined.

Attribute	Data Type	Required?	Default	Description
				• name$.recordingsize = Size of the recording in bytes, or undefined if no audio is collected. • name$.recordingduration = Duration of the recording in ms, or undefined if no audio is collected. • name$.utterance = Raw string of recognized words.
slot	identifier	no	N/A	Name of the grammar slot used to populate the variable. If not specified, the value is the field item variable name.
type	CDATA	no	N/A	Name of an internal grammar. Input can be through voice or DTMF. Valid values are: • boolean = Recognizes negative or positive responses. Accepts the following parameters: • boolean = Default boolean type (no parameters specified). A key press of 1 is the affirmative answer, and a key press of 2 is the negative answer. • boolean?y=d = DTMF grammar that treats the key press d as an affirmative answer. • boolean?n=e = DTMF grammar that treats the key press e as a negative answer. • currency = Recognizes currency amounts (e.g., $56.78). The default currency is euro number, which recognizes real numbers (e.g., 3.14 or 77). Currently not supported.

Attribute	Data Type	Required?	Default	Description
				• date = Recognizes date formats (e.g., *YYYYMMDD* 20020317). Currently not supported. • digits = Recognizes the digits 0 through 9. • digits?minlength=n = String of at least n digits. • digits?maxlength=m = String of at most m digits. • digits?length=p = String of exactly p digits. • phone = Recognizes telephone numbers (e.g., 800-555-1234). Currently not supported. • time = Recognizes hours and minutes (e.g., 10:45). Currently not supported. • none = Always recognizes an empty match and returns immediately. As with any input item, it will trigger the line connection, the play of prompts queue, etc. Basic syntax is <field type="none"/>.

Parents

<form>

Children

<audio>, <catch>, <enumerate>, <error>, <filled>, <grammar>, <help>, <link>, <noinput>, <nomatch>, <option>, <prompt>, <property>, <value>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <field name="sevendigits" type="digits?length=7" modal="true">
            <prompt>Say or type a seven digit number</prompt>
        </field>
        <field name="answer" type="boolean" modal="true">
            <prompt>Answer yes or no</prompt>
        </field>
        <filled namelist="sevendigits answer">
            <prompt>
                Your number was
                <say-as interpret-as="vxml:digits">
                    <value expr="sevendigits"/>
                </say-as>
```

```
                <break time="80ms"/>
                    and you answered <value expr="answer"/>
                <break time="80ms"/> Bye
            </prompt>
        </filled>
    </form>
</vxml>
```

See Also

<form>, <initial>

<filled>

Action to perform after a form input item is filled.

Syntax

```
<filled
    mode = "{all | any}"
    namelist = "list of identifiers"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
mode	identifier	no	all	Determines when the <filled> element is executed. Valid values: - all = <filled> element is executed when all of the fields are filled. - any = <filled> element is executed when any of the fields are filled. If the <filled> element is specified at the form-item level, the mode attribute cannot be specified in the <filled> element.
namelist	list of identifiers	no	N/A	Space-separated list of input item variables. Control items are not permitted in this list. When omitted, the namelist defaults to the names of all of the form's input items. If the <filled> tag is specified at the form-item level, the namelist attribute cannot be specified in the <filled> tag.

Details

The <filled> element can be a child of:

- An input item, in which case the actions are executed when that item is filled.
- The <form> element, in which case the actions are executed as determined by the mode attribute.

Parents

<field>, <form>, <object>, <record>, <subdialog>, <transfer>

Children

<assign>, <audio>, <clear>, <data>, <disconnect>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <reprompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <field name="userID" type="digits?length=8">
            <prompt>What is your user ID?</prompt>
                <noinput>
                    Please say or key in your eight digit user ID
                    <reprompt/>
                </noinput>
        </field>
        <field name="pin" type="digits?length=4">
            <prompt>What is your pin number?</prompt>
                <noinput>
                    Please say or key in your four digit pin number
                    <reprompt/>
                </noinput>
        <!--
            A field level <filled> - does not specify mode or namelist.
        -->
                <filled>
                    <if cond="pin == 9999">
                        Entry denied - please retry
                        <clear namelist="pin"/>
                    <elseif cond="pin == 0000"/>
                        Invalid pin - please re try
                        <clear namelist="pin"/>
                    <else/>
                        You entered <value expr="pin"/>
                    </if>
                </filled>
        </field>
    <!--
        A form level <filled> - executes when both fields are filled:
    -->
        <filled namelist="userID pin" mode="all">
            <submit next="validate.jsp"/>
        </filled>
    </form>
</vxml>
```

See Also

<field>, <form>, <object>, <record>, <subdialog>, <transfer>

<foreach>

Iterates through an ECMAScript array, executing the contained executable content for each item in the array. Typically, <foreach> is used to concatenate prompts dynamically.

Syntax

```
<foreach
    array="CDATA"
    item="identifier">
    <!-- do something -->
</foreach>
```

Attributes

Attribute	Data Type	Required?	Default	Description
array	CDATA	yes	N/A	ECMAScript expression that must evaluate to an array; otherwise, an error.semantic event is thrown.
item	identifier	yes	N/A	Variable that stores each array item upon each iteration of the loop. A new variable is declared if it is not already defined within the parent's scope.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>, <prompt>

Children

<assign>, <audio>, <clear>, <data>, <disconnect>, <else>, <elseif>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <reprompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```
foreach

<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <script>
        var movies = new Array(3);
        movies[0] = new Object();
        movies[0].audio = "godfather.wav";
        movies[0].tts = "the godfather";
        movies[1] = new Object();
        movies[1].audio = "high fidelity.wav";
        movies[1].tts = "high fidelity";
        movies[2] = new Object();
        movies[2].audio = "raiders.wav";
        movies[2].tts = "raiders of the lost ark";
    </script>
    <form id="pick movie">
        <field name="movie">
            <grammar type="application/srgs+xml" src="movie_names.grxml"/>
                <prompt>
                    <audio src="prelist.wav">
```

```
                      When you hear the name of the movie you want,
                            just say it.
                </audio>
                <foreach item="thePrompt" array="movies">
                    <audio expr="thePrompt.audio">
                        <value expr="thePrompt.tts"/>
                    </audio>
                    <break time="300ms"/>
                </foreach>
            </prompt>
        </field>
    </form>
</vxml>
```

See Also

<if>, <prompt>

<form>

Dialog that defines an interaction for collecting values for a set of field item variables.

Syntax

```
<form
   id = "identifier"
   scope = "{document | dialog}"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
id	identifier	no	N/A	Unique name of the form. Having a unique name facilitates the transfer of control from form to form.
scope	{document | dialog}	no	dialog	Scope for the grammar contained in the form. Valid values: • dialog = Grammar is only active in the current form. • document = Grammar is active throughout the current document.

Details

A form can contain the following items:

- Input items filled by user utterances.
- Control items that control form execution.
- Event handlers.
- Dialog-scoped variables.
- Filled items, which specify the action to take after a field is input.
- Properties.

Parents

<vxml>

Children

<block>, <catch>, <data>, <error>, <field>, <filled>, <grammar>, <help>, <initial>, <link>, <noinput>, <nomatch>, <object>, <property>, <record>, <script>, <subdialog>, <transfer>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="creditCard">
        <field name="cardtype">
            <prompt>Which credit card type ?</prompt>
            <grammar type="application/srgs+xml" root="cardtype">
                <rule id="cardtype">
                    <one-of>
                        <item>american express <tag>amex</tag></item>
                        <item>master card <tag>mc</tag></item>
                        <item>visa <tag>visa</tag></item>
                    </one-of>
                </rule>
            </grammar>

            <noinput count="1">
                Sorry I didn't hear you
                <reprompt/>
            </noinput>
            <noinput count="2">
                Sorry still didn't hear you
                <reprompt/>
            </noinput>
            <nomatch count="1">
                <prompt>
                    Sorry I didn't understand please repeat the card type
                </prompt>
            </nomatch>
            <nomatch count="2">
                <prompt>
                    I still don't understand please select either american
                        express or master card or visa
                </prompt>
            </nomatch>

            <filled>
                <prompt>
                    You selected <value expr="cardtype"/>
                </prompt>
            </filled>
        </field>
```

```
    </form>
</vxml>
```

See Also

<menu>

<goto>

Transfers execution to the specified URI.

Syntax

```
<goto
    expr = "CDATA"
    expritem = "CDATA"
    fetchaudio = "URI"
    fetchhint = "{prefetch | safe}"
    fetchtimeout = "CDATA"
    maxage = "CDATA"
    maxstale = "CDATA"
    next = "URI"
    nextitem = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
expr	CDATA	no	N/A	ECMAScript expression that evaluates to the next dialog URI fragment.
expritem	CDATA	no	N/A	ECMAScript expression that evaluates to the next form item to transition to.
fetchaudio	URI	no	N/A	URI of an audio resource to play while the next document resource is being fetched. If the fetchaudio attribute is not set, no audio is played during the fetch. The audio is terminated after the next document is fetched.

Attribute	Data Type	Required?	Default	Description
fetchhint	{prefetch | safe}	no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: • prefetch = Fetch the resource when the page is loaded. • safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Time interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh.
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.
next	URI	no	N/A	The URI fragment of the next dialog, or URI of the next document to load.
nextitem	CDATA	no	N/A	The next form item in the current form to transition to.

Details

The destination of a <goto> operation can be:

- An item in the current form (e.g., a named block)
- A dialog in the current document (e.g., a named form)
- A new document

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```xml
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="form1">
        <block name="block1">
            <prompt>Currently in block one</prompt>
            <goto nextitem="block3"/>
        </block>
        <block name="block2">
            <prompt>Now in block two</prompt>
        </block>
        <block name="block3">
            <prompt>Now in block three</prompt>
            <goto next="#form2"/>
        </block>
    </form>
    <form id="form2">
        <block name="block4">
        <prompt>Now in form two</prompt>
        </block>
    </form>
</vxml>
```

See Also

<submit>

<grammar>

Specifies the valid spoken utterances and corresponding string values returned in response to the utterances.

Syntax

<grammar
   fetchhint = "{prefetch | safe}"
   fetchtimeout = "CDATA"
   maxage = "CDATA"
   maxstale = "CDATA"
   mode = "{voice | dtmf}"
   root = "CDATA"
   scope = "{document | dialog}"
   src = "URI"
   srcexpr = "CDATA"
   tag-format = "CDATA"
   type = "CDATA"
   version = "CDATA"
   weight = "CDATA"
   xml:base = "URI"
  xml:lang = "locale identifier"
   xmlns = "CDATA"
   xmlns:xsi = "CDATA"

```
    xsi:schemalocation = "CDATA"
/>
```

Attributes

None of the <grammar> attributes are required.

Attribute	Data Type	Required?	Default	Description
fetchhint	{prefetch | safe}	no	None.	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: - prefetch = Fetch the resource when the page is loaded. - safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Time interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh.
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.

Attribute	Data Type	Required?	Default	Description
mode	{voice | dtmf}	no	voice	Input type this grammar recognizes. Valid values: • voice = Grammar recognizes voice input. • dtmf = Grammar recognizes DTMF input.
root	CDATA	no	N/A	Root rule for the document.
scope	{document | dialog}	no	dialog	Scope for the grammar contained in the form. Valid values: • document = Grammar is active throughout the current document. • dialog = Grammar is active only in the current form.
src	URI	no	N/A	URI of the grammar file.
srcexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI of the grammar file.
tag-format	CDATA	no	semantic/ 1.0-literals	Tag format of the grammar. **Note:** DTMF grammar only supports semantic/1.0-literals.

Attribute	Data Type	Required?	Default	Description
type	CDATA	no	N/A	MIME type of the grammar. If omitted, the VoiceXML Interpreter attempts to determine the type dynamically. The type attribute can also specify the character encoding of the grammar, as follows: `type="application/srgs+xml; charset=UTF-8".`
version	CDATA	no	1.0	Currently not supported.
weight	CDATA	no	N/A	Currently not supported.
xml:lang	locale identifier	no	N/A	Language and locale of the document, using an identifier compliant with RFC 1766.
xml:base	URI	no	N/A	Base URI from which relative URIs in the grammar are resolved.
xmlns	CDATA	no	N/A	(Stand-alone SRGS XML documents only) Designated namespace for the SRGS XML format.
xmlns:xsi	CDATA	no	N/A	(Stand-alone SRGS XML documents only) Used with the xsi:schemalocation attribute to indicate the location of the schema for the SRGS XML namespace.
xsi:schemalocation	CDATA	no	N/A	Used with xmlns:xsi attribute to indicate the location of the schema for the SRGS XML namespace.

Details

The XML form of the Speech Recognition Grammar Specification (SRGS) is also supported in VoiceXML 2.1 applications. Grammars for DTMF are currently sent to the PowerMedia XMS local DTMF recognizer. The local DTMF recognizer only supports semantics/1.0-literals tag-format.

Parents

<choice>, <field>, <form>, <link>, <option>, <record>, <transfer>

Children

<lexicon>, <meta>, <rule>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="creditCard">
        <field name="cardtype">
            <prompt>Which credit card type ?</prompt>
            <grammar type="application/srgs+xml" root="cardtype">
                <rule id="cardtype">
                    <one-of>
                        <item>american express <tag>amex</tag></item>
                        <item>master card <tag>mc</tag></item>
                        <item>visa <tag>visa</tag></item>
                    </one-of>
                </rule>
            </grammar>
            <noinput count="1">
                Sorry I didn't hear you
                <reprompt/>
            </noinput>
            <noinput count="2">
                Sorry still didn't hear you
                <reprompt/>
            </noinput>
            <nomatch count="1">
                <prompt>
                    Sorry I didn't understand please repeat the card type
                </prompt>
            </nomatch>
            <nomatch count="2">
                <prompt>
                  I still don't understand please select either american
                    express or master card or visa
                </prompt>
            </nomatch>
            <filled namelist="cardtype">
                <prompt>
                    You selected <value expr="cardtype"/>
                </prompt>
            </filled>
        </field>
    </form>
</vxml>
```

See Also

<form>

<help>

Catches the help event.

Syntax

```
<help
    cond = "CDATA"
    count = "integer"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the <help> element to execute.
count	integer	no	N/A	Numerical occurrence of the help event, such as 2 for the second occurrence. This allows you to handle different occurrences of the help event in different ways.

Details

The <help> element is shorthand for <catch event="help">.

Parents

<field>, <form>, <initial>, <menu>, <object>, <record>, <subdialog>, <transfer>, <vxml>

Children

<assign>, <audio>, <clear>, <data>, <disconnect>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <reprompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <catch event="terminate">
        Caught terminate
        <goto next="#formExit"/>
    </catch>
    <help>
        <audio src="../../audio/help.wav" maxage="0"/>
            Help requested
        <goto next="#formEntry"/>
    </help>
    <error>
        Caught error
        <goto next="#formExit"/>
    </error>
    <var name="bCondition" expr="false"/>
    <form id="formEntry">
        <catch event="customEvent">
            Form scope custom handler
            Throwing terminate
            <throw event="terminate"/>
        </catch>
        <catch event="terminate" cond="bCondition">
            This handler will not be executed due to the condition
```

```
            </catch>
            <block>
                Throwing custom event
                <throw event="customEvent"/>
            </block>
    </form>
    <form id="formExit">
        <block>
            Goodbye
            <exit/>
        </block>
    </form>
</vxml>
```

See Also

<catch>

<if>

Used in if..elseif..else conditional logic.

Syntax

```
<if cond="CDATA">
    <!-- do something -->
<elseif cond="CDATA"/>
  <!-- do something else -->
<elseif cond="CDATA"/>
  <!-- do something else -->
<else/>
    <!-- do something else -->
</if>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	yes	N/A	Boolean expression that must evaluate to ECMAScript true for the <if> element to execute.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

<assign>, <audio>, <clear>, <data>, <disconnect>, <else>, <elseif>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <reprompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <field name="pin" type="digits?length=4">
            <prompt>What is your pin number?</prompt>
            <noinput>
                Please say or key in your four digit pin number
            <reprompt/>
```

```
            </noinput>
            <filled>
                <if cond="pin == 9999">
                    Entry denied - please re try
                    <clear namelist="pin"/>
                <elseif cond="pin == 0000"/>
                    Invalid pin - please retry
                    <clear namelist="pin"/>
                <else/>
                    You entered <value expr="pin"/>
                </if>
            </filled>
        </field>
    </form>
</vxml>
```

See Also

<else>, <elseif>

<initial>

Control item that specifies the entry logic for a mixed initiative form.

Syntax

```
<initial
    cond = "CDATA"
    expr = "PCDATA"
    name = "identifier"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the <initial> element to execute.
expr	PCDATA	no	ECMAScript undefined	Initial value of the form item variable.
name	identifier	no	N/A	Name of the form item variable. This must be a legal ECMAScript identifier.

Details

The <initial> element prompts the user, expecting an utterance matching a form-level grammar to fill multiple fields. When a field is filled by an utterance, all <initial> items in the current form are set. They are not set again unless explicitly cleared.

The <initial> element requires a form-level grammar. Field grammars are not active while the <initial> element executes.

Parents

<form>

Children

<audio>, <catch>, <enumerate>, <error>, <help>, <link>, <noinput>, <nomatch>, <prompt>, <property>, <value>

Example

```xml
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="weather_info">
        <grammar src="../../grammar/weatherService.grxml" maxage="0"/>
        <block>
            <prompt bargein="false">
                Welcome to the weather information service.
                <audio src="http://www.online-ads.example/wis.wav"/>
                    Vox pilot develop anywhere deploy everywhere
            </prompt>
        </block>
        <initial name="start">
            <prompt>
                For what city and country would you like the weather?
            </prompt>
            <help>
                Please say the name of the city and country for which you
                    would like a weather report.
            </help>
        </initial>
        <field name="country">
            <prompt>What country?</prompt>
                <help>
                    Please speak the country for which you want the weather.
                </help>
        </field>
        <field name="city">
            <prompt>
                Please say the city in <value expr="country"/>
                    for which you want the weather.
            </prompt>
            <help>
                Please speak the city for which you want the weather.
            </help>
        </field>
        <field name="go ahead" type="boolean" modal="true">
            <prompt>
                Do you want to hear the weather
                    for <value expr="city"/>
                    in <value expr="country"/>
            </prompt>
            <filled>
                <if cond="go_ahead">
                    <!-- Servlet example, does not exist -->
                    <submit next="/servlet/weather" method="post"
                        namelist="city country" fetchtimeout="45s" />
                </if>
                <clear namelist="start city country go_ahead"/>
            </filled>
        </field>
    </form>
</vxml>
weatherService.grxml:
<?xml version="1.0"?>
<grammar version="1.0" root="weatherservice"
    xmlns="http://www.w3.org/2001/06/grammar"
    xml:lang="en-gb">
    <rule id="weatherservice">
        <one-of>
            <item>dublin <tag>dublin</tag></item>
            <item>cork <tag>cork</tag></item>
            <item>amsterdam <tag>amsterdam</tag></item>
        </one-of>
```

```
        </rule>
</grammar>
```

See Also

<field>, <form>

<item>

Defines a valid utterance match within an XML grammar.

Syntax

```
<item
    repeat = "integer"
    repeat-prob = "CDATA"
    weight = "CDATA"
/>
```

Attributes

None of the <item> attributes are required.

Attribute	Data Type	Default	Description
repeat	integer	N/A	Integer range that specifies how many times the item can be repeated.
repeat-prob	CDATA	N/A	Currently not supported.
weight	CDATA	N/A	Probability weighting of this item.

Parents

<item>, <one-of>, <rule>

Children

<item>, <one-of>, <ruleref>, <tag>

Example

```
<?xml version= "1.0"?>
<grammar xmlns="http://www.w3.org/2001/06/grammar"
    xml:lang="en-GB" root="ruleid">
    <rule id="ruleid" scope="public">
        <one-of>
            <item>
                irish
                <tag><fieldone "irish"></tag>
            </item>
            <item weight="0.1">
                english
                <tag><fieldone "english"></tag>
            </item>
            <item weight="0.9">
                french
                <tag><fieldone "french"></tag>
            </item>
        </one-of>
    </rule>
</grammar>
```

<lexicon>

Location of a pronunciation lexicon document that contains mappings of words to substitute words, and mappings of words to phoneme sequences.

Syntax

```
<lexicon
    type = "CDATA"
    uri = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
type	CDATA	no	N/A	Media type of the pronunciation lexicon document.
uri	CDATA	yes	N/A	URI of the pronunciation lexicon document.

Details

The SSML specification does not define a default lexicon format. Until such a format is defined, the format supported is a subset of a W3C XML Lexicon Working Draft specification (with media type application/x-lexicon).

The root element in a lexicon document is the <lexicon> element. This can include an alphabet attribute, which specifies the alphabet used in phonetic sequences contained in the document, if any. The SSML <phoneme> element defines the valid values for the <lexicon> element. If the alphabet is unspecified or unrecognized, the SSML Processor uses the International Phonetic Alphabet (IPA).

The lexicon document can contain zero or more <lexeme> elements. Each <lexeme> element contains child text that specifies the target word or phrase to be substituted, and either a <sounds-like> element or a <phoneme> element:

- The <sounds-like> element uses child text to specify the substitute word or phrase with which the target word or phrase is to be replaced. This is equivalent to the SSML <sub> element.

- The <phoneme> element uses the ph attribute to specify a sequence of phonemes with which the target word or phrase is to be replaced.

Parents

<grammar>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
    <block>
            <prompt>
```

```
              <lexicon uri="DublinAsLondon.lexicon" type="application/x-lexicon"/>
          </prompt>
       </block>
    </form>
</vxml>
```

See Also

<phoneme>, <sub>

<link>

Specifies one or more grammars, which are scoped to the element containing the link.

Syntax

```
<link
   dtmf = "CDATA"
   event = "CDATA"
   eventexpr = "CDATA"
   expr = "CDATA"
   fetchaudio = "URI"
   fetchint = "{prefetch | safe}"
   fetchtimeout = "CDATA"
  next = "URI"
   scope = "{document | dialog}"
   src = "URI"
   tag-format = "CDATA"
   type = "CDATA"
   version = "CDATA"
   weight = "CDATA"
   xml:lang = "locale identifier"
   xmlns = "CDATA"
   xmlns:xsi = "CDATA"
   xsi:schemalocation = "CDATA"
/>
```

Attributes

None of the attributes are required.

Attribute	Data Type	Required?	Default	Description
dtmf	CDATA	no	N/A	DTMF key used to validate a successful grammar match.
event	CDATA	no	N/A	Event that can be raised to the application.
eventexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the event which can be raised to the application.

Attribute	Data Type	Required?	Default	Description
expr	CDATA	no	N/A	ECMAScript value that evaluates to the target URI.
fetchaudio	URI	no	N/A	URI of an audio resource to play while the XML data is fetched. If the fetchaudio attribute is not set, no audio is played during the fetch. If the audio is playing after the XML document is fetched, the audio terminates.
fetchhint	{prefetch | safe}	no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: • prefetch = Fetch the resource when the page is loaded. • safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Time interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
next	URI	no	N/A	URI of the dialog (form or menu) or document to transition to when the user input matches one of the grammars specified by <link>.

Attribute	Data Type	Required?	Default	Description
scope	{dialog | document}	no	dialog	Scope for the grammar contained in the form. Valid values: • dialog = Grammar is active only in the current form. • document = Grammar is active throughout the current document.
src	URI	no	N/A	URI of the grammar file.
tag-format	CDATA	no	Nuance	Tag format of the grammar. Currently the only supported format is of type Nuance.
type	CDATA	no	N/A	MIME type of the grammar. If omitted, the VoiceXML Interpreter attempts to determine the type dynamically. The type attribute can also specify the character encoding of the grammar, as follows: `type="application/srgs+xml; charset=UTF-8"`
version	CDATA	no	N/A	SRGS version of this grammar. Currently not supported.
weight	CDATA	no	N/A	Currently not supported.
xml:lang	locale identifier	no	N/A	Language and locale of the document using an identifier compliant with RFC 1766.

Attribute	Data Type	Required?	Default	Description
xmlns	CDATA	no	N/A	Designated namespace for the SRGS XML format. Applies to stand-alone SRGS XML documents only.
xml:xsi	CDATA	no	N/A	Used with xsi:schemalocation to indicate the location of the schema for the SRGS XML namespace. Applies to stand-alone SRGS XML documents only.
xmlschemalocation	CDATA	no	N/A	Used with xmlns:xsi to indicate the location of the schema for the SRGS XML namespace. Applies to stand-alone SRGS XML documents only.

Details

When one of the grammars specified by <link> is matched, the link is activated and does one of the following:

- Transitions to a new dialog or document using the value of the next or expr attributes.
- Raises an event using the value of the event or eventexpr attributes.

Parents

<field>, <form>, <initial>, <vxml>

Children

<grammar>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <link next="#linkTest">
        <grammar type="application/srgs+xml" root="Repeat">
            <rule id="Repeat">
                <item>repeat</item>
            </rule>
        </grammar>
    </link>
    <catch event="quit">
        Goodbye
```

```
        <exit/>
    </catch>
    <form id="init">
        <field name="link" >
            <prompt>
                Say repeat to execute the link
                    <break time="90ms"/>
                Or say quit to exit
                    <break time="200ms"/>
            </prompt>
        <link event="quit">

         <grammar type="application/srgs+xml" root="Quit">
            <rule id="Quit">
                <item>quit</item>
            </rule>
         </grammar>

        </link>
        </field>
        <catch event="noinput nomatch">
            <reprompt/>
        </catch>
    </form>
    <form id="linkTest">
        <block>
            This is the link test
            <goto next="#init"/>
        </block>
    </form>
</vxml>
```

See Also

<form>

<log>

Generates a log message to use for performance monitoring or debugging purposes.

Syntax

```
<log
    expr = "CDATA"
    label = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
expr	CDATA	no	N/A	ECMAScript expression that evaluates to the label.
label	CDATA	no	N/A	String that specifies the purpose or severity of the log message.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```xml
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="form1">
        <block name="block1">
            <log expr="'Entered block1'" label="LOG-1"/>
            <prompt>Currently in block one</prompt>
            <goto nextitem="block2"/>
        </block>
        <block name="block2">
            <log expr="'Entered block2'" label="LOG-2"/>
            <prompt>Now in block two</prompt>
            <goto next="#form2"/>
        </block>
    </form>
    <form id="form2">
        <block name="block3">
            <var name="x" expr="28"/>
            <log label="LOG-3" expr="'Entered block3'">
                Variable x has value: <value expr="x"/>
            </log>
            <prompt>Now in form two</prompt>
        </block>
    </form>
</vxml>
```

<mark>

Places a marker into a text or element sequence.

Syntax

```
<mark
   name = "CDATA"
   nameexpr = "CDATA"
```

`/>`

Attributes

Attribute	Data Type	Required?	Default	Description
name	CDATA	no	N/A	Name associated with this mark.
nameexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the mark name.

Details

Although the VoiceXML Interpreter supports this element, it does not currently support mark notifications, so the lastresult$.markname and lastresult$.marktime variables (and the corresponding form item variables) are never set.

Parents

<audio>, <choice>, <emphasis>, <enumerate>, <p>, <prompt>, <prosody>, <s>, <voice>

Children

None.

Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <var name="played_ad" expr="false"/>
    <form>
        <field name="jacket">
            <prompt>
                <mark name="order_start"/>
                    Fleece jackets are available in a wide array of colors,
                        including red, green, purple, brown, blue, and black. You
                        won't find a better price.
                <mark name="order_end"/>
                <break time="500ms"/>
                    Say the name of the color you want to view.
                        For example, say purple.
            </prompt>
        <grammar type="application/srgs+xml" src="jackets.grxml"/>
            <filled>
                <if cond="typeof(jacket$.markname) == 'string' &amp;&amp;
                    (jacket$.markname=='order end' ||
                    (jacket$.markname=='order_start' &amp;&amp;
                      jacket$.marktime &gt;= 5000))">
                  <assign jacket="played_order" expr="true"/>
                <else/>
                    <assign jacket="played_order" expr="false"/>
                </if>
            </filled>
        </field>
    </form>
</vxml>
```

<media>

Plays the media file at the specified location.

Syntax

```
<media
   src = "URI"
   expr = "CDATA"
   fetchhint = "{prefetch | safe}"
   fetchtimeout = "CDATA"
   maxage = "CDATA"
   maxstale = "CDATA"
   offset = "CDATA"
   offsetexpr = "CDATA"
   type = "CDATA"
   dlgc:videotype = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
expr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI (location) of the media file. Programmable Media Platform proprietary.
fetchhint	{prefetch | safe}	no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: • prefetch = Fetch the resource when the page is loaded. • safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Time interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh.
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.

Attribute	Data Type	Required?	Default	Description
offset	CDATA	no	N/A	The offset (in s or ms) at which to play the media file. This attribute is supported for: • HTTP or local files • HTTP or local audio-only files
offsetexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the offset. This can be used in conjunction with the lastresult$.bargeintime variable to let the application play a media file from the point at which the user barged in, and not have to start from the beginning again. This attribute is especially useful for long media files.
src	CDATA	no	N/A	URI of the media file.
type	CDATA	no	N/A	Optionally specifies the media type of the requested resource when the resource type cannot be derived from a VXML recording or the specified resource URI file extension. When a recording file is specified in the attribute of a <record> element, it is not recommended to specify the type attribute because the type will be automatically taken from the recording data. If the type attribute is specified, the specified type attribute will take precedence.

Attribute	Data Type	Required?	Default	Description
dlgc:videotype	CDATA	no	N/A	Optionally specifies the media type of the requested resource when the resource type cannot be derived from a VXML recording or the specified resource URI file extension. Valid values: • video/x-vid • video/3gpp When a recording file is specified in the attribute of a \<record\> element, it is not recommended to specify the dlgc:videotype attribute because the dlgc:videotype will be automatically taken from the recording data. If the dlgc:videotype attribute is specified, the specified dlgc:videotype attribute will take precedence.

Details

In this implementation, the \<audio\> and \<media\> elements are completely equivalent, with the exception that \<media\> can specify a type of application/ssml+xml, omitting the src and expr attributes and enclosing a complete SSML document as its child content. This content is not treated as alternate content.

Parents

\<audio\>, \<block\>, \<catch\>, \<choice\>, \<enumerate\>, \<error\>, \<field\>, \<filled\>, \<foreach\>, \<help\>, \<if\>, \<initial\>, \<media\>, \<menu\>, \<noinput\>, \<nomatch\>, \<object\>, \<prompt\>, \<record\>, \<subdialog\>, \<transfer\>

Children

\<audio\>, \<break\>, \<desc\>, \<emphasis\>, \<enumerate\>, \<p\>, \<mark\>, \<media\>, \<phoneme\>, \<s\>, \<prosody\>, \<say-as\>, \<speak\>, \<sub\>, \<value\>, \<voice\>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <prompt>
                <media type="application/ssml+xml">
                    <speak version="1.0" xml:lang="en-GB">
                        Welcome to your Video Mail account!
                            You have <value expr="num messages"/> messages.
                    </speak>
                </media>
            </prompt>
        </block>
```

```
    </form>
</vxml>
```

See Also

<audio>, <prompt>

<menu>

Allows the user to specify a choice from a list of options.

Syntax

```
<menu
   accept = "{exact | approximate}"
   dtmf = "{true | false}"
   id = "identifier"
   scope = "{document | dialog}"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
accept	{exact | approximate}	no	exact	Determines how a choice phrase is activated. Valid values: • exact = Utterances must match the entire choice phrase to activate the choice. • approximate = Utterances can match a sub-phrase of the choice phrase to activate the choice. Each choice can override this setting.

Attribute	Data Type	Required?	Default	Description
dtmf	boolean	no	no	Assigns implicit DTMF elements to the choices contained in the menu. Valid values: - true = Assigns implicit DTMF elements to any <choice> element that does not specify a DTMF element. - false = Does not assign implicit DTMF elements to <choice> elements.
id	identifier	no	N/A	Unique name of the menu. This facilitates the transfer of control from dialog to dialog.
scope	{document | dialog}	no	dialog	Scope for the grammar contained in the menu. Valid values: - document = Grammar is active throughout the current document. - dialog = Grammar is active only in the current menu.

Parents

<vxml>

Children

<audio>, <catch>, <choice>, <enumerate>, <error>, <help>, <noinput>, <nomatch>, <prompt>, <property>, <script>, <value>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <catch event="quitEvent">
        Goodbye
        <exit/>
    </catch>
    <!--
    With dtmf="true", the DTMF sequence for each
    choice will be assigned automatically: 1, 2 and 3
```

```
    -->
    <menu id="main" dtmf="true">
        <prompt>What do you want to eat, apple or orange?</prompt>
        <choice next="#formApple"> apple </choice>
        <choice expr="'#' + 'formOrange'"> orange </choice>
        <choice event="quitEvent"> quit </choice>
        <nomatch>Please say one of <enumerate/></nomatch>
    </menu>
    <form id="formApple">
        <block>
            You chose apple
            <exit/>
        </block>
    </form>
    <form id="formOrange">
        <block>
            You chose orange
            <exit/>
        </block>
    </form>
</vxml>
```

See Also

<choice>, <form>

<meta>

Specifies meta information about the current VoiceXML, SRGS XML, or SSML document.

Syntax

```
<meta
    content = "CDATA"
    name = "identifier"
    http-equiv = "identifier"
/>
```

Attributes

None of the <meta> attributes are required.

Attribute	Data Type	Required?	Default	Description
content	CDATA	yes	N/A	Value of the metadata property.
http-equiv	identifier	no	N/A	Name of the metadata property.
name	identifier	no	N/A	Name of an HTTP response header.

Parents

<vxml>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <meta name="author" content=""/>
    <meta name="maintainer" content=""/>
    <form>
        <block>
            <prompt>A simple vxml test</prompt>
        </block>
    </form>
</vxml>
```

<metadata>

Container in which information about the current VoiceXML, SRGS XML, or SSML document is placed, using a metadata schema.

Syntax

Attributes

None.

Parents

<grammar>, <prompt>, <vxml>

Children

None.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2001/vxml
    http://www.w3.org/TR/voicexml20/vxml.xsd">
    <metadata>
        <rdf:RDF
            xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:rdfs = "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
            xmlns:dc = "http://purl.org/metadata/dublin_core#">
            <!-- Metadata about the VoiceXML document -->
            <rdf:Description about="http://www.example.com/meta.vxml"
                dc:Title="Directory Enquiry Service"
                dc:Description="Directory Enquiry Service for London in VoiceXML"
                dc:Publisher="W3C"
                dc:Language="en"
                dc:Date="2002-02-12"
                dc:Rights="Copyright 2002 John Smith"
                dc:Format="application/voicexml+xml" >
                <dc:Creator>
                    <rdf:Seq ID="CreatorsAlphabeticalBySurname">
                        <rdf:li>Jackie Crystal</rdf:li>
                        <rdf:li>William Lee</rdf:li>
                    </rdf:Seq>
                </dc:Creator>
            </rdf:Description>
        </rdf:RDF>
    </metadata>
    <form>
        <block>
            <prompt>Hello</prompt>
        </block>
```

```
        </form>
</vxml>
```

<noinput>

Catches the noinput event.

Syntax

```
<noinput
    cond = "CDATA"
    count = "integer"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the <noinput> element to execute.
count	integer	no	N/A	Numerical occurrence of the noinput event, such as 2 for the second occurrence. This allows the application to handle different occurrences of the noinput event in different ways.

Details

This element is shorthand for <catch event="noinput">.

Parents

<field>, <form>, <initial>, <menu>, <object>, <record>, <subdialog>, <transfer>, <vxml>

Children

<audio>, <assign>, <clear>, <data>, <disconnect>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <reprompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="creditCard">
        <field name="cardtype">
            <prompt>Which credit card type ?</prompt>
            <grammar type="application/srgs+xml" root="cardtype">
                <rule id="cardtype">
                    <one-of>
                        <item>american express <tag>amex</tag></item>
                        <item>master card <tag>mc</tag></item>
                        <item>visa <tag>visa</tag></item>
                    </one-of>
                </rule>
            </grammar>
            <noinput count="1">
```

```
                    Sorry I didn't hear you
                    <reprompt/>
            </noinput>
            <noinput count="2">
                    Sorry still didn't hear you
                    <reprompt/>
            </noinput>
            <nomatch count="1">
                    <prompt>
                        Sorry I didn't understand please repeat the card type
                    </prompt>
            </nomatch>
            <nomatch count="2">
                    <prompt>
                        I still don't understand please select either american
                            express or master card or visa
                    </prompt>
            </nomatch>
            <filled namelist="cardtype">
                    <prompt>
                        You selected <value expr="cardtype"/>
                    </prompt>
            </filled>
        </field>
    </form>
</vxml>
```

See Also

<catch>

<nomatch>

Catches the nomatch event.

Syntax

```
<nomatch
   cond = "CDATA"
   count = "integer"
/>
```

Attributes

Attribute	Data Type	Default	Description
cond	CDATA	N/A	Boolean expression that must evaluate to ECMAScript true for the <nomatch> element to execute.
count	integer	N/A	Numerical occurrence of the nomatch event, such as 2 for the second occurrence. This allows you to handle different occurrences of the nomatch event in different ways.

Details

This element is shorthand for <catch event="nomatch">.

Parents

<field>, <form>, <initial>, <menu>, <object>, <record>, <subdialog>, <transfer>, <vxml>

Children

<audio>, <assign>, <clear>, <data>, <disconnect>, <enumerate>, <exit>, <foreach>, <goto>, <if>, <log>, <prompt>, <reprompt>, <return>, <script>, <submit>, <throw>, <value>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="creditCard">
        <field name="cardtype">
            <prompt>Which credit card type ?</prompt>
            <grammar type="application/srgs+xml" root="cardtype">
                <rule id="cardtype">
                    <one-of>
                        <item>american express <tag>amex</tag></item>
                        <item>master card <tag>mc</tag></item>
                        <item>visa <tag>visa</tag></item>
                    </one-of>
                </rule>
            </grammar>
            <noinput count="1">
                Sorry I didn't hear you
                <reprompt/>
            </noinput>
            <noinput count="2">
                Sorry still didn't hear you
                <reprompt/>
            </noinput>
            <nomatch count="1">
                <prompt>
                    Sorry I didn't understand please repeat the card type
                </prompt>
            </nomatch>
            <nomatch count="2">
                <prompt>
                    I still don't understand please select either american express
                        or master card or visa
                </prompt>
            </nomatch>
            <filled namelist="cardtype">
                <prompt>
                    You selected <value expr="cardtype"/>
                </prompt>
            </filled>
        </field>
    </form>
</vxml>
```

See Also

<catch>

<object>

Invokes a platform-specific object.

Syntax

<object
    archive = "CDATA"

```
    classid = "CDATA"
    codebase = "CDATA"
    codetype = "CDATA"
    cond = "CDATA"
    data = "CDATA"
    expr = "CDATA"
    fetchaudio = "URI"
    fetchhint = "{prefetch | safe}"
    fetchtimeout = "CDATA"
    name = "identifier"
    type = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
archive	CDATA	no	N/A	Currently not supported.
classid	CDATA	no	N/A	URI specifying the location of the object's implementation.
codebase	CDATA	no	N/A	Currently not supported.
codetype	CDATA	no	N/A	Currently not supported.
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the <object> element to execute.
data	CDATA	no	N/A	Currently not supported.
expr	CDATA	no	N/A	Initial value of the form item variable.
fetchaudio	URI	no	N/A	Currently not supported.
fetchhint	{prefetch | safe}	no	N/A	Currently not supported.
fetchtimeout	CDATA	no	N/A	Currently not supported.
name	identifier	no	ECMAScript undefined	Name of the object variable. This must be a legal ECMAScript identifier.
type	CDATA	no	N/A	Currently not supported.

Parents

<form>

Children

<audio>, <catch>, <enumerate>, <error> , <filled>, <noinput>, <help>, <nomatch>, <param>, <prompt>, <property>, <value>

Example

```
<?xml version="1.0" encoding="utf-8"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml" xml:lang="en-GB">
    <form id="playDTMF">
        <object name="objDtmf1" classid="no specific-platform object is currently available">
            <param name="dtmfstring" value="1"/>
        </object>
    </form>
</vxml>
```

See Also

<form>, <param>

<one-of>

Enables the grammar to be constructed with a series of alternate phrases or rule expansions, each of which is contained within an <item> element.

Syntax

```
<one-of
   xml:lang = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
xml:lang	CDATA	no	N/A	Language used in the grammar token, conforming to RFC 3066.

Details

The <one-of> element specifies a disjunction in an SRGS XML grammar, allowing the grammar to specify a list of alternatives using multiple <item> elements.

Parents

<item>, <rule>

Children

<item>

Example

```
<grammar xmlns="http://www.w3.org/2001/06/grammar"
    xml:lang="en-GB" root="rule1">
    <rule id="rule1" scope="public">
        <one-of>
            <item>
                irish
```

```
                    <tag><field1 "irish"></tag>
            </item>
            <item>
                english
                <tag><field1 "english"></tag>
            </item>
        </one-of>
    </rule>
</grammar>
```

<option>

Generates a grammar for a simple response to field options.

Syntax

```
<option
    accept = "{exact | approximate}"
    dtmf = "CDATA"
    value = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
accept	{approximate | exact}	no	exact	Determines how the option is activated. Valid values: • approximate = Utterance must match a sub-phrase to activate this option. • exact = Utterance must match the entire choice phrase to activate this option.
dtmf	CDATA	no	N/A	DTMF sequence for this option.
value	CDATA	no	N/A	String to assign to the field item variable when a user selects this option by speech or DTMF.

Details

If the dtmf attribute is unspecified, the option cannot be matched using DTMF.

If the value attribute is unspecified, the VoiceXML Interpreter assigns default data to the option as follows:

- Uses the CDATA content of the <option> element with leading and trailing white space removed.
- If CDATA content does not exist, then the VoiceXML Interpreter uses the DTMF sequence.
- If neither the CDATA content nor a DTMF sequence is specified, then the default assignment is undefined and the field's form item variable is not filled.

Parents

<field>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <meta name="author" content=""/>
    <meta name="maintainer" content=""/>
    <form id="options">
        <catch event="nomatch noinput">
            <prompt> Please select an option. </prompt>
            <reprompt/>
        </catch>
        <field name="myOption">
            <prompt>
                Please select from the following list: <enumerate/>
            </prompt>
            <option value="inquiry"> billing inquiry </option>
            <option value="statement"> last statement </option>
            <option value="balance"> current balance </option>
            <option value="support"> customer support </option>
            <filled>
                <prompt>You chose <value expr="myOption"/> </prompt>
            </filled>
        </field>
    </form>
</vxml>
```

See Also

<choice>, <field>

<p>

Identifies the enclosed text as a paragraph containing zero or more sentences.

Syntax

```
<p
  xml:lang = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
xml:lang	CDATA	no	N/A	RFC 1766 compliant identifier used to locate the text-to-speech (TTS) engine that will render the enclosed text.

Parents

<audio>, <enumerate>, <prosody>, <prompt>, <voice>

Children

<audio>, <break>, <emphasis>, <enumerate>, <mark>, <phoneme>, <prosody>, <s>, <say-as>, <sub>, <value>, <voice>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
   <form>
        <block>
            <prompt>
                <p>
                     <s>This is the first sentence of the paragraph.</s>
                     <s>Here's another sentence.</s>
                </p>
            </prompt>
        </block>
    </form>
</vxml>
```

See Also

<s>

<par>

With the <media> element, provides parallel playback of separate media sources of differing types.

Syntax

<par></par>

Attributes

None.

Details

The <par> element, allows simultaneous playback of multiple media resources during a video call. In this implementation, you can simultaneously play back a single audio and a single video stream from separate sources.

Specify the media streams using child <media> elements. Exactly two such elements must be specified: one specifying an audio-only source and the other specifying a video-only source. If one of the elements references an audio-video source, playback does not occur.

For the media source, the <media> element must reference an external source using the src or expr attribute.

If one of the sources cannot be played (e.g., because of an error fetching a file), the other is still rendered. If a <par> element executes while in a voice call, only the audio stream plays.

The following restrictions apply to the use of <par>:

- Exactly two <media> elements must be specified. If this condition is violated, an error.semantic is raised.

- A <prompt> element containing <par> cannot contain any other child elements. To queue audio before or after the parallel playback, place the audio in a separate <prompt> element. A <prompt> element must be either parallel or sequential, and not a combination of both.

- In the current implementation, it is not possible to reference more than one consecutive audio or video source in a single <media> element during parallel playback. For example, if the audio <media> element contains inline SSML that uses text-to-speech, that SSML cannot also contain an <audio> or <media> element referencing an external source.

- The <property> element is currently not supported when it is used as a child of <par>.

Parents

<prompt>

Children

<audio>, <media>, <property>

Example

The following example shows how the <par> element can use the src attribute of the <media> element to reference an audio source and a video source:

```xml
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <prompt>
                <par>
                    <media src="http://example.org/media.3gp"/>  <!-- Audio part -->
                    <media src="http://example.org/media.3gp"/>  <!-- Video part -->
                </par>
            </prompt>
        </block>
    </form>
</vxml>
```

See Also

<audio>, <media>, <prompt>

<param>

Specifies a parameter to pass to an object or subdialog.

Syntax

```
<param
    expr = "CDATA"
    name = "identifier"
    type = "CDATA"
    value = "CDATA"
    valuetype = "{data | ref}"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
expr	CDATA	no	ECMAScript undefined	Expression that computes the value associated with the name attribute.
name	identifier	no	N/A	Parameter name.
type	CDATA	no	N/A	Currently not supported.
value	CDATA	no	N/A	Literal string value associated with the name attribute.
valuetype	CDATA	no	data	Currently not supported.

Parents

<object>, <subdialog>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <var name="defaultName"/>
    <form>
        <block>
            Calling the sub dialog
            <assign name="document.defaultName" expr="'Fred'"/>
        </block>
        <subdialog name="subResult" src="#mySubDialog">
            <param name="firstName" expr="document.defaultName"/>
            <param name="lastName" value="Smith"/>
        </subdialog>
        <block>
            After the call to the sub dialog the name is
            <value expr="subResult.firstName"/>
            <value expr="subResult.lastName"/>
        </block>
    </form>
    <form id="mySubDialog">
        <var name="firstName" expr="'Billy'"/>
        <var name="lastName"/>
        <block>
            In the sub dialog
            <value expr="firstName"/>
```

```
            <value expr="lastName"/>
            <return namelist="firstName lastName"/>
        </block>
    </form>
</vxml>
```

<phoneme>

Phonetic pronunciation for the contained text.

Syntax

```
<phoneme
    alphabet = "CDATA"
    ph = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
alphabet	CDATA	no	N/A	Alphabet to use for the phonetic string.
ph	CDATA	yes	N/A	Phonetic string for the enclosed text.

Parents

<audio>, <emphasis>, <enumerate>, <p>, <prompt>, <prosody>, <s>, <voice>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <prompt>
                <phoneme alphabet="ipa" ph="t&#x252;m&#x251;to&#x28A;">
                    tomato
                </phoneme>
            </prompt>
        </block>
    </form>
</vxml>
```

See Also

<lexicon>, <prosody>, <say-as>, <voice>

<prompt>

Controls the output of text-to-speech (TTS) and audio sources.

Syntax

```
<prompt
    bargein = "{true | false}"
    bargeintype = "{speech | hotword}"
    cond = "CDATA"
    count = "integer"
    timeout = "integer"
    version = "CDATA"
    xml:base = "CDATA"
    xml:lang = "CDATA"
/>
```

Attributes

None of the <prompt> attributes are required.

Attribute	Data Type	Required?	Default	Description
bargein	boolean	no	yes	Indicates whether a user can interrupt the prompt. This attribute overrides a bargein previously set with the <property> element.
bargeintype	{speech | hotword}	no	speech	Type of bargein that can interrupt this prompt. This attribute overrides a bargein type previously set with the <property> element.
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the prompt to be played.
count	integer	no	1	A number that allows you to emit different prompts if the user does something repeatedly.
timeout	integer	no	N/A	Silence time, in seconds, after which a noinput event is raised. This attribute overrides a timeout set previously by the <property> element.
version	CDATA	no	N/A	Currently not supported.
xml:base	CDATA	no	N/A	Currently not supported.

Attribute	Data Type	Required?	Default	Description
xml:lang	CDATA	no	N/A	RFC 1766 compliant identifier used to locate the text-to-speech (TTS) engine used to render text prompts.

Details

SSML elements provide a hint as to how the enclosed text should be spoken. Whether or not a certain element has an effect depends on the capabilities of the TTS engine currently in use. In some cases, only certain attributes of an element have an effect, and even then, the level of control may not be as high as is described in the specification.

For a list of elements and attributes supported by the current set of TTS engines, see SSML Support.

Parents

<block>, <catch>, <error>, <help>, <field>, <filled>, <foreach>, <help>, <if>, <menu>, <initial>, <noinput>, <nomatch>, <object>, <record>, <subdialog>, <transfer>

Children

<audio>, <break>, <emphasis>, <enumerate>, <lexicon>, <mark>, <meta>, <metadata>, <p>, <foreach>, <phoneme>, <prosody>, <s>, <say-as>, <sub>, <value>, <voice>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="french">
    <block>
            <prompt xml:lang="fr-FR">
                bonjour, bienvenue au service vocal.
            </prompt>
            <goto next="#english"/>
        </block>
    </form>
    <form id="english">
        <block>
            <prompt xml:lang="en-GB">
                hello, welcome to the voice portal.
            </prompt>
            <goto next="#spanish"/>
        </block>
    </form>
    <form id="spanish">
        <block>
            <prompt xml:lang="es-ES">
                buenos días, bienvenida sobre el servicio vocal.
            </prompt>
        </block>
    </form>
</vxml>
```

See Also

<audio>

<property>

Sets a property value.

Syntax

```
<property
    name = "identifier"
    value = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
name	identifier	yes	N/A	Name of the property to set.
value	CDATA	yes	N/A	Value to assign to the property.

Details

Properties control platform behavior, including caching, timeouts, and so forth. For a list of supported properties, see Application Properties.

Parents

<field>, <form>, <initial>, <menu>, <object>, <record>, <subdialog>, <transfer>, <vxml>

Children

None.

Example

```xml
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <property name="audiomaxage" value="0"/>
    <property name="audiofetchhint" value="safe"/>
    <property name="confidence" value="0.75"/>
    <property name="bargein" value="true"/>
    <form id="weather_info">
        <grammar src="../../grammar/weatherService.grxml"/>
            <block>
                <prompt bargein="false">
                    Welcome to the weather information service.
                        Barge in is enabled but not here
                </prompt>
            </block>
            <initial name="start">
                <property name="timeout" value="5s"/>
                <prompt>
                    For what city and country would you like the weather?
                </prompt>
                <help>
                    Please say the name of the city and country
                        for which you would like a weather report.
                </help>
            </initial>
            <field name="country">
                <prompt>What country?</prompt>
                <help>
                    Please speak the country for which you want the weather.
                </help>
```

```
            </field>
            <field name="city">
                <prompt>
                    Please say the city in <value expr="country"/>
                        for which you want the weather.
                </prompt>
                <help>
                    Please speak the city for which you want the weather.
                </help>
            </field>
            <field name="go_ahead" type="boolean" modal="true">
                <prompt>
                    Do you want to hear the weather
                        for <value expr="city$.utterance"/>
                        in <value expr="country$.utterance"/>
                </prompt>
                <filled>
                    <if cond="go_ahead">
                        <!-- Servlet example, does not exist -->
                        <submit next="/servlet/weather" method="post"
                            namelist="city country" fetchtimeout="45s" />
                    </if>
                    <clear namelist="start city country go_ahead"/>
                </filled>
            </field>
        </form>
</vxml>
weatherService.grxml:
<?xml version="1.0"?>
    <grammar version="1.0" root="weatherservice"
        xmlns="http://www.w3.org/2001/06/grammar"
        xml:lang="en-gb">
        <rule id="weatherservice">
            <one-of>
                <item>dublin <tag>dublin</tag></item>
                <item>cork <tag>cork</tag></item>
                <item>amsterdam <tag>amsterdam</tag></item>
            </one-of>
        </rule>
    </grammar>
```

<prosody>

Specifies the pitch, duration, speaking rate, and volume for the enclosed speech output.

Syntax

```
<prosody
   contour = "CDATA"
   duration = "CDATA"
   pitch = "CDATA"
   range = "CDATA"
   rate = "CDATA"
   volume = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
contour	CDATA	no	N/A	Pitch contour of the speech output, formatted as a value pair: • The first value is a percentage of the period of the contained text (a number followed by %). • The second value is the value of the pitch attribute.
duration	CDATA	no	N/A	Time duration for reading the speech output, in seconds (s) or milliseconds (ms). For example, 5s or 3500ms.
pitch	CDATA	no	N/A	Level and intensity of the speech output. Valid values: • A number followed by Hz. • A relative change, as compared to the default pitch. • One of the following values: high, low, medium, x-high, x-low, default. A relative change is expressed using the plus sign (+) or minus sign (-), followed by Hz (hertz) or st (semitones). It can also be expressed as a percentage change preceded by an optional + or -. The TTS engine determines the default pitch.

Attribute	Data Type	Required?	Default	Description
range	CDATA	no	N/A	Pitch range of the speech output. Valid values: • A number followed by Hz, where higher values increase the pitch range. • A relative change, as compared to the default range. • One of the following values: high, low, medium, x-high, x-low, default. A relative change is expressed using the plus sign (+) or minus sign (-), followed by a number, followed by Hz (hertz) or st (semitones). It can also be expressed as a percentage change preceded by an optional + or -. The TTS engine determines the default range.
rate	CDATA	no	N/A	Speaking rate of the speech output. Valid values: • A number followed by Hz. • A relative change, as compared to the default rate. • One of the following values: high, low, medium, x-high, x-low, default. A relative change is expressed as a number that acts as a multiplier of the default rate. Thus, a value of 2 means the rate should be twice the default rate; a value of 0.5 means the rate should be half the default rate. The TTS engine determines the default rate.

Attribute	Data Type	Required?	Default	Description
volume	CDATA	no	N/A	Volume of the speech output. Valid values: • A number between 0.0 and 100.0, where higher values are louder. • A relative change, as compared to the default volume. • One of the following values: x-loud, loud, medium, soft, x-soft, silent, default. A relative change is expressed using the plus sign (+) or minus sign (-), followed by a number. It can also be expressed as a percentage change preceded by an optional + or -. The TTS engine determines the default volume.

Parents

<audio>, <emphasis>, <enumerate>, <p>, <prompt>, <prosody>, <s>, <voice>

Children

<audio>, <break>, <emphasis>, <enumerate>, <mark>, <p>, <phoneme>, <prosody>, <s>, <say-as>, <sub>, <value>, <voice>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <prompt>
                The price of XYZ is
                <prosody volume="loud" rate="0.5">
                <say-as interpret-as="vxml:currency">$45</say-as>
                </prosody>
            </prompt>
        </block>
    </form>
</vxml>
```

See Also

<emphasis>, <say-as>, <voice>

<record>

An input item that collects media input and stores the recording in the form item variable.

Syntax

```
<record
    beep = "{true | false}"
    cond = "CDATA"
    dlgc:dest = "CDATA"
    dlgc:destexpr = "CDATA"
    dlgc:lang = "locale identifier"
    dlgc:videoname = "identifier"
    dlgc:videotype = "CDATA"
    dtmfterm = "{true | false}"
    expr = "CDATA"
    finalsilence = "CDATA"
    maxtime = "CDATA"
    modal = "{true | false}"
    name = "identifier"
    type = "CDATA"
/>
```

The following declaration must be used to enable the dlgc domain name:

```
xmlns:dlgc="http://www.dialogic.com/xmlns"
```

Attributes

Attribute	Data Type	Required?	Default	Description
beep	boolean	no	False	Indicates whether to emit a tone prior to recording. Valid values: • true = Emit a tone. • false = Do not emit a tone.
cond	CDATA	no	N/A	Specifies a Boolean expression that must evaluate to ECMAScript true for the record to execute.

Attribute	Data Type	Required?	Default	Description
dlgc:dest	CDATA	no	N/A	Specifies a URI where the recording will be stored.
				Note 1: If using absolute paths for the dlgc:dest/destexpr attribute, the "Allow absolute paths" option must be checked on the Media page of the WebGUI.
				Note 2: The recorded file is not automatically deleted when the channel is released.
				Note 3: The recorded file content is not loaded or fetched into a VXML memory variable. The variable references the file URI of the recording (the same content as dlgc:dest/destexpr).
				Note 4: The <submit> of the VXML memory variable that usually contains the recording shall submit the URI of recording when dlgc:dest/destexpr was used to record.
				Note 5: When playing the record variable, the URI of the recording will be used internally to play the file.

Attribute	Data Type	Required?	Default	Description
dlgc:destexpr	CDATA	no	N/A	Specifies a URI where the recording will be stored, and evaluates a javascript expression to dynamically construct that URI. **Note 1:** If using absolute paths for the dlgc:dest/destexpr attribute, the "Allow absolute paths" option must be checked on the Media page of the WebGUI. **Note 2:** The recorded file is not automatically deleted when the channel is released. **Note 3:** The recorded file content is not loaded or fetched into a VXML memory variable. The variable references the file URI of the recording (the same content as dlgc:dest/destexpr). **Note 4:** The \<submit\> of the VXML memory variable that usually contains the recording shall submit the URI of recording when dlgc:dest/destexpr was used to record. **Note 5:** When playing the record variable, the URI of the recording will be used internally to play the file.
dlgc:lang	locale identifier	no	N/A	Specifies the language and locale of the document. **Note:** dlgc:lang is ignored if using an absolute path.

Attribute	Data Type	Required?	Default	Description
dlgc:videoname	identifier	no	N/A	Specify the VXML name for the video recording.
dlgc:videotype	CDATA	no	N/A	Define the video type. Valid values: • video/x-vid • video/3gpp Attribute is optional. If not present, use the API default.
dtmfterm	boolean	no	True	Indicates whether to terminate the recording with a DTMF key press. Valid values: • true = Terminate the recording with a DTMF key press. • false = Do not terminate the recording with a DTMF key press. **Note:** Associated tones, if any, are not part of the recording.
expr	CDATA	no	N/A	Initial value for the form item variable.
finalsilence	integer	no	N/A	Length of the silence that indicates the end of the recording, in seconds (s) or milliseconds (ms).
maxtime	CDATA	no	300s	Maximum length of the recording, in seconds (s) or milliseconds (ms). If not specified, the default is 300 seconds.
modal	boolean	no	N/A	Currently not supported.
name	identifier	no	N/A	Name of the input item variable that holds the recording.

Attribute	Data Type	Required?	Default	Description
type	CDATA	no	audio/ x-wav	Media format of the resulting recording. This can be one of the audio file formats specified by the VXML specification. Valid values: • audio/x-wav = WAV (RIFF header). Valid codecs are L8, L16, mulaw, alaw, or native. If the native codec is not supported by the container, L16 will be used. Valid rates are 800, 11025 (L8 and L16), or 16000 (L8 and L16) (e.g., audio/x-wav; codec=L8 rate=11025). • audio/x-aud = Dialogic proprietary. Valid codecs are L8, L16, mulaw, alaw, amr, amr-wb, or native. If the native codec is not supported by the container, L16 will be used. Valid rates are 800, 11025 (L8 and L16), or 16000 (L8 and L16. Valid modes are 0 through 7 (default is 7) for amr and 0 through 8 (default is 8) for amr-wb (e.g., audio/x-wav; codec=L8 rate=16000 mode=7).

Attribute	Data Type	Required?	Default	Description
				• audio/basic = Raw (headerless) 8kHz 8-bit mu-law PCM (G.711). • audio/x-alaw-basic = Raw (headerless) 8kHz 8-bit A-law PCM (G.711). • audio/L8 = Raw (headerless) 8-bit linear PCM. Valid rates are 8000 (default), 11025, or 16000 (e.g., audio/L8; rate=8000). • audio/L16 = Raw (headerless) 16-bit linear PCM. Valid rates are 8000 (default), 11025, or 16000 (e.g., audio/L16; rate=8000). • audio/amr = Valid modes are 0 through 7 (e.g., audio/amr; mode=7). The default is 7. • audio/amr-wb = Valid modes are 0 through 8 (e.g., audio/amr-wb;mode=8). The default is 8. • audio/3gpp = AMR or AMR-WB. For AMR, the codec is amr and the valid modes are 0 through 7 (e.g., audio/3gpp; codec=amr mode=7).

Attribute	Data Type	Required?	Default	Description
				For AMR-WB, the codec is amr-wb and the valid modes are 0 through 8 (e.g., audio/3gpp; codec=amr-wb mode=8). The default is 8.
				If the empty string is specified, the system will use the file extension to determine the type. The default type is equivalent to audio/x-wav; codec=L8 rate=8000.

Details

If there is no dlgc:dest or dlgc:destexpr destination attribute specified, the system will create a temporary recording file and delete it at the channel release.

Parents

<form>

Children

<audio>, <enumerate>, <help>, <catch>, <error>, <filled>, <grammar>, <noinput>, <nomatch>, <prompt>, <property>, <value>

Example

```
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <record  name="msg" beep="true" maxtime="10s" finalsilence="4000ms"
            dtmfterm="true" type="audio/x-wav">
            <prompt timeout="5s">
                Record a message after the beep.
            </prompt>
            <noinput>
                I didn't hear anything, please try again.
            </noinput>
        </record>
        <field name="confirm">
            <grammar type="application/srgs+xml" src="/grammars/boolean.grxml"/>
            <prompt>
                Your message is <audio expr="msg"/>.
            </prompt>
            <prompt>
                To keep it, say yes. To discard it, say no.
            </prompt>
            <filled>
                <if cond="confirm">
                    <submit next="save_message.pl" enctype="multipart/form-data"
                        method="post" namelist="msg"/>
                </if>
                <clear/>
            </filled>
```

```
        </field>
    </form>
</vxml>
```

See Also

<object>, <subdialog>

<reprompt>

After a catch is executed, causes normal prompt processing to occur (queuing and counter increments) instead of suppressing prompt processing.

Syntax

<reprompt/>

Attributes

None.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="creditCard">
        <field name="cardtype">
            <prompt>Which credit card type ?</prompt>
                <grammar type="application/srgs+xml" root="cardtype">
                    <rule id="cardtype">
                        <one-of>
                            <item>american express <tag>amex</tag></item>
                            <item>master card <tag>mc</tag></item>
                            <item>visa <tag>visa</tag></item>
                        </one-of>
                    </rule>
                </grammar>
                <noinput count="1">
                    Sorry I didn't hear you
                    <reprompt/>
                </noinput>
                <noinput count="2">
                    Sorry still didn't hear you
                    <reprompt/>
                </noinput>
                <nomatch count="1">
                    <prompt>
                        Sorry I didn't understand please repeat the card type
                    </prompt>
                </nomatch>
                <nomatch count="2">
                    <prompt>
                        I still don't understand.
                        please select either american express or master card or visa
                    </prompt>
                </nomatch>
            <filled namelist="cardtype">
                <prompt>
                    You selected <value expr="cardtype"/>
                </prompt>
```

```
            </filled>
        </field>
    </form>
</vxml>
```

<return>

Ends execution in a subdialog. Returns control and optionally, data, to the caller of the subdialog.

Syntax

```
<return
    event = "CDATA"
    eventexpr = "CDATA"
    message = "CDATA"
    messageexpr = "CDATA"
    namelist = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
event	CDATA	no	N/A	Event to raise returning to the caller.
eventexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the event to raise.
message	CDATA	no	N/A	String that provides additional information about the raised event. The message is available as the ECMAScript variable _message in the <catch> element that handles the event.
messageexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the message.
namelist	CDATA	no	N/A	Space-separated list of variables to return to the caller.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```xml
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <var name="defaultName"/>
    <form>
        <block>
            Calling the sub dialog
            <assign name="document.defaultName" expr="'Fred'"/>
        </block>
            <subdialog name="subResult" src="#mySubDialog">
                <param name="firstName" expr="document.defaultName"/>
                <param name="lastName" value="Smith"/>
            </subdialog>
        <block>
            After the call to the sub dialog the name is
            <value expr="subResult.firstName"/>
            <value expr="subResult.lastName"/>
        </block>
    </form>
    <form id="mySubDialog">
        <var name="firstName" expr="'Billy'"/>
        <var name="lastName"/>
        <block>
            In the sub dialog
            <value expr="firstName"/>
            <value expr="lastName"/>
            <return namelist="firstName lastName"/>
        </block>
    </form>
</vxml>
```

See Also

<subdialog>

<rule>

Defines the named rule expansion of an XML grammar.

Syntax

```
<rule
   id = "identifier"
   scope = "{private | public}"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
id	identifier	yes	N/A	Identifier by which to reference this rule.
scope	"{private | public}"	no	private	Scope of the rule. Valid values: • private • public

Parents

Children

<example>, <item>, <one-of>, <ruleref>, <tag>, <token>

Example

```
<?xml version= "1.0"?>
    <grammar xmlns="http://www.w3.org/2001/06/grammar"
        xml:lang="en-GB" root="rule1">
        <rule id="rule1" scope="public">
            <one-of>
                <item>irish</item>
            </one-of>
        </rule>
    </grammar>
```

<ruleref>

Allows the specification of an existing rule for inclusion within the current grammar.

Syntax

```
<ruleref
    special = "{NULL | VOID | GARBAGE}"
    type = "CDATA"
    uri = "URI"
    lang-list = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
special	NULL| VOID| GARBAGE	no	N/A	Currently there is only support for tag-format of type "Nuance".
type	CDATA	no	application/srgs +xml	Type of the referenced rule. Only application/srgs+xml is supported.
uri	URI	no	N/A	URI of the grammar to reference.
xml:lang	CDATA	no	N/A	Language used in the referenced rule.

Parents

<item>, <rule>

Children

None.

Example

```
<?xml version= "1.0"?>
    <grammar xmlns="http://www.w3.org/2001/06/grammar"
```

```
        xml:lang="en-GB" root="rule1">
        <rule id="rule1">
            <one-of>
                <item>
                    <ruleref uri="http://myserver.com/"/>
                </item>
            </one-of>
        </rule>
    </grammar>
```

<s>

Identifies the enclosed text as a sentence.

Syntax

```
<s
   xml:lang = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
xml:lang	CDATA	no	N/A	RFC 1766 compliant identifier used to locate the text-to-speech (TTS) engine that renders the enclosed text.

Parents

<audio>, <enumerate>, <p>, <prompt>, <prosody>, <voice>

Children

<audio>, <break>, <emphasis>, <enumerate>, <mark>, <phoneme>, <prosody>, <say-as>, <sub>, <value>, <voice>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <var name="x" expr="12345"/>
                <prompt>
                    <p>
                        <s>This is the first sentence of the paragraph.</s>
                        <s>Here's another sentence.</s>
                    </p>
                </prompt>
        </block>
    </form>
</vxml>
```

See Also

<p>

<say-as>

Specifies how a word or phrase is spoken.

This feature can be processed either by a remote mrcp engine or by the local builtin engine. The processing engine is selected by setting the com.dlgc.sayas property.

The builtin say-as is converted internally by a phrase server to a list of files to play according to the requested interpret-as and format fields.

The builtin engine supports the following languages:

- en-US
- sp-SP
- zh-CN

The builtin and mrcp format fields could be different for the same interpret-as value according to the remote engine specification.

Property	Description
com.dlgc.sayas	builtin (default) mrcp

The property can be inserted in the <vxml> scope in order to apply to all scripts or in the <form> scope to apply only to that context. The <form> scope overrides <vxml> scope.

Syntax

```
<say-as
    detail = "PCDATA"
    format = "PCDATA"
    interpret-as = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
detail	PCDATA	no	N/A	Level of detail to read aloud or render. This field is ignored with the builtin engine.
format	PCDATA	no	N/A	Provides precise formatting information for the contained text for content types with ambiguous formats. • mrcp = According to the remote engine specification. • builtin = See interpret-as attribute.
interpret-as	CDATA	yes	N/A	Style of a given text format. Valid values:

Attribute	Data Type	Required?	Default	Description
				VXML standard: • vxml:digits = Speak contained text as digits (e.g., one two three instead of one hundred and twenty three). No builtin format values. • vxml:number = Contained text is a number. Builtin format values are *ord* for ordinal (e.g., 1=first) and *crd* for cardinal (e.g., 1=one). Default is *crd*. • vxml:date = Contained text is a date YYYYMMDD. Builtin format values for dates are *ymd* (default), *dmy*, *mdy*, *ym*, *my*, and *y*. • vxml:time = Contained text is a time of day in the type HHMMx where x is *a* for am, *p* for pm, and *h* for 24h. Builtin accepts also HHMM (without x) in 24h with format *t12* or *t24* (default). • vxml:currency = Contained text is a currency amount. Text should be XXXmm.nn where XXX is the standardized 3-letters currency unit. For builtin case, only USD is allowed and if format contains the currency unit (USD), the text must contain only the amount. • vxml:phone = Contained text is a telephone number. • vxml:boolean = Speak contained text as an affirmative or negative phrase appropriate to the current locale (not supported with builtin engine).

Attribute	Data Type	Required?	Default	Description
				Builtin enhancement: • string = Pronounce contained text as individual characters. • duration = Contained text is a number in seconds and spoken using hour, minute, and second units. • month = Contained text is a number MM from 1 to 12 and spoken as the month name. • silence = Plays a specified period of silence as indicated by the duration value in milliseconds (e.g., silence:duration=1000). • weekday = Contained text is a number representing the day in the week (e.g., 1=Sunday, .., 7=Saturday).

Parents

<audio>, <emphasis>, <enumerate>, <p>, <prompt>, <prosody>, <s>, <voice>

Children

<value>

Example

In the following example, the text that results from evaluating the expression is spoken in the style specified by the enclosing <say-as> element.

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
<property name="com.dlgc.sayas" value="builtin"/>
    <form>
    <block>
            <var name="x" expr="12345"/>
            <prompt>
              date <say-as interpret-as="string">RTE</say-as>
               date <say-as interpret-as="vxml:date" format="mdy">20140425</say-as>
               digits <say-as interpret-as="vxml:digits"> <value expr="x"/></say-as>
               number <say-as interpret-as="vxml:number">54321</say-as>
               phone <say-as interpret-as="vxml:phone">123-456-7890</say-as>
               currency <say-as interpret-as="vxml:currency">USD123.56</say-as>
               currency <say-as interpret-as="vxml:currency" format="USD">34123.56</say-as>
               time <say-as interpret-as="vxml:time">0816p</say-as>
               time <say-as interpret-as="vxml:time" format="t12">2016</say-as>
            </prompt>
        </block>
```

```
    </form>
</vxml>
```

See Also

<emphasis>, <prosody>, <voice>

<script>

Allows the specification of a block of client-side scripting language code.

Syntax

```
<script
    charset = "CDATA"
    fetchtimeout = "CDATA"
    fetchhint = "{prefetch | safe}"
    maxage = "CDATA"
    maxstale = "CDATA"
    src = "URI"
    srcexpr = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
charset	CDATA	no	N/A	Character encoding of the script designated by the src attribute.
fetchtimeout	CDATA	no	N/A	Time interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
fetchhint		no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: • prefetch = Fetch the resource when the page is loaded. • safe = Fetch the resource when it is specifically called by the application.

Attribute	Data Type	Required?	Default	Description
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh.
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.
src	URI	no	N/A	URI of the script file.
srcexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI of the script file.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <form>, <help>, <if>, <noinput>, <nomatch>, <vxml>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <script>

        var d = new Date();
        var hours = d.getHours();
        var minutes = d.getMinutes();
        var seconds = d.getSeconds();

    </script>
    <form>
        <block>
            The time is <value expr="hours"/> hours,
            <value expr="minutes"/> minutes, and
            <value expr="seconds"/> seconds.
        </block>
    </form>
</vxml>
```

See Also

<assign>, <var>

<speak>

Top-level element required in a standalone SSML document.

Syntax

```
<speak
   version = "CDATA"
   xml:base = "URI"
   xml:lang = "CDATA"
   xmlns = "CDATA"
   xmlns:xsi = "CDATA"
   xsi:schemalocation = "CDATA"
    />
```

Attributes

Attribute	Data Type	Required?	Default	Description
version	CDATA	yes	N/A	VoiceXML version.
xml:base	URI	no	N/A	Base URI, used to resolve relative URIs in the document.
xml:lang	CDATA	no	N/A	Language variant of the document.
xmlns	CDATA	no	N/A	Designated namespace for VoiceXML.
xmlns:xsi	CDATA	no	N/A	Used with the xsi:schemalocation attribute to indicate the location of the schema for the VoiceXML namespace.
xsi:schemalocation	CDATA	no	N/A	Used with the xmlns:xsi attribute to indicate the location of the schema for the VoiceXML namespace.

Details

In-line SSML fragments are permitted in VoiceXML 2.0 documents. In this case, the <prompt> element is used in place of the <speak> element, allowing the same SSML child elements and attributes, except xmlns, xmlns:xsi, and xsi:schemaLocation.

When only SSML fragments are used, the generated VXML automatically generates SSML with a <speak> element containing the following attributes: version="1.0", the current xml:lang, and xmlns="http://www.w3.org/2001/10/synthesis".

The <speak> element describes how the enclosed text should be spoken. Whether or not a certain element has an effect depends on the capabilities of the TTS engine in use. In some cases, only certain attributes of an element have an effect, and even then, the level of control may not be as high as described in the specification.

The <speak> element is allowed only if it is the single child element of a <media> element and if that <media> element contains the attribute type="application/ssml+xml".

For information about the elements and attributes supported by the current set of TTS engines, see SSML Support.

Parents

<media>

Children

<audio>, <break>, <emphasis>, <mark>, <phoneme>, <p>, <prosody>, <s>, <say-as>, <sub>, <voice>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <var name="num_messages" expr="3"/>
    <form>
        <block>
            <media type="application/ssml+xml">
                <speak version="1.0" xml:lang="en-GB"
                xmlns="http://www.w3.org/2001/10/synthesis">
                    Welcome to your Video Mail account!
                    You have <value expr="num_messages"/> messages.
                </speak>
            </media>
        </block>
    </form>
</vxml>
```

See Also

<prompt>

<sub>

Indicates that the specified text replaces the contained text for pronunciation. This allows a document to contain both a spoken and written form.

Syntax

<sub
  alias = "CDATA"
/>

Attributes

Attribute	Data Type	Required?	Default	Description
alias	CDATA	yes	N/A	String to substitute for the enclosed string.

Parents

<audio>, <emphasis>, <enumerate>, <p>, <prompt>, <prosody>, <s>, <voice>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <var name="x" expr="12345"/>
            <prompt>
                <sub alias="Speech Synthesis Markup Language">SSML</sub>
            </prompt>
        </block>
    </form>
</vxml>
```

See Also

<lexicon>

<subdialog>

Invokes a dialog in a new execution context. Control returns to the calling element after the called subdialog executes a return.

Syntax

```
<subdialog
    cond = "CDATA"
    enctype = "CDATA"
    expr = "CDATA"
    fetchaudio = "URI"
    fetchtimeout = "CDATA"
    fetchhint = "{prefetch | safe}"
    maxage = "CDATA"
    maxstale = "CDATA"
    method = "{get | post}"
    name = "identifier"
    namelist = "list of identifiers"
    src = "URI"
    srcexpr = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the <subdialog> element to execute.

Attribute	Data Type	Required?	Default	Description
enctype	CDATA	no	application /x-www- form- urlencoded	MIME encoding type to use when submitting data to the application server while fetching the subdialog. To submit an audio file that results from a record operation, use enctype="multipart/form-data".
expr	CDATA	no	N/A	Initial value of the form item variable.
fetchaudio	URI	no	N/A	URI of an audio resource to play while the XML data is fetched. If the fetchaudio attribute is not set, no audio is played during the fetch. If the audio is playing after the XML document is fetched, the audio terminates.
fetchhint	{prefetch | safe}	no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: - prefetch = Fetch the resource when the page is loaded. - safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Time interval to wait for the content to be returned before raising an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh.

Attribute	Data Type	Required?	Default	Description
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.
method	{get | post}	no	get	HTTP request method. Valid values: • get = Execute an HTTP get. • post = Execute an HTTP post.
name	identifier	no	N/A	Result returned from the subdialog as an ECMAScript object whose properties are defined in the namelist attribute of the <return> element.
namelist	list of identifiers	no	N/A	Space-separated list of variables sent to the subdialog.
src	URI	no	N/A	URI of the subdialog.
screxpr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI of the subdialog.

Parents

<form>

Children

<audio>, <catch>, <enumerate>, <error>, <filled>, <help>, <noinput>, <nomatch>, <param>, <prompt>, <property>, <value>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <var name="defaultName"/>
    <form>
        <block>
            Calling the sub dialog
            <assign name="document.defaultName" expr="'Fred'"/>
        </block>
        <subdialog name="subResult" src="#mySubDialog">
            <param name="firstName" expr="document.defaultName"/>
            <param name="lastName" value="Smith"/>
        </subdialog>
        <block>
```

```
          After the call to the sub dialog the name is
          <value expr="subResult.firstName"/>
          <value expr="subResult.lastName"/>
      </block>
   </form>
   <form id="mySubDialog">
      <var name="firstName" expr="'Billy'"/>
      <var name="lastName"/>
      <block>
          In the sub dialog
          <value expr="firstName"/>
          <value expr="lastName"/>
          <return namelist="firstName lastName"/>
      </block>
   </form>
</vxml>
```

See Also

<form>, <return>

<submit>

Performs an HTTP GET or POST with optional variables.

Syntax

```
<submit
   enctype = "CDATA"
   expr = "CDATA"
   fetchaudio = "URI"
   fetchtimeout = "CDATA"
   fetchhint = "{prefetch | safe}"
   maxage = "CDATA"
   maxstale = "CDATA"
   method = "{get | post}"
   namelist = "list of identifiers"
   next = "URI"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
enctype	CDATA	no	application/ x-www- form- urlencoded	MIME encoding type of the submitted document. A MIME type of multipart/form-data must be specified for uploading to the server the audio file that results from a record operation.
expr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI of the HTTP request.

Attribute	Data Type	Required?	Default	Description
fetchaudio	URI	no	N/A	URI of an audio resource to play while the XML data is fetched. If the fetchaudio attribute is not set, no audio is played during the fetch. If the audio is playing after the XML document is fetched, the audio terminates.
fetchhint	{prefetch | safe}	no	N/A	Specifies when the VoiceXML Interpreter context should retrieve content from the server. Valid values: • prefetch = Fetch the resource when the page is loaded. • safe = Fetch the resource when it is specifically called by the application.
fetchtimeout	CDATA	no	N/A	Time interval to wait for the content to be returned before throwing an error.badfetch event. Use s for seconds (e.g., 1s) and ms for milliseconds (e.g., 1ms).
maxage	CDATA	no	N/A	Maximum acceptable age, in seconds, of a resource being fetched from the cache. Setting maxage to 0 means that a cached version is never considered fresh.
maxstale	CDATA	no	N/A	Maximum acceptable staleness, in seconds, of the resource being fetched, if the fetched resource is cached and expired.

Attribute	Data Type	Required?	Default	Description
method	{get | post}	no	get	HTTP request method. Valid values: • get = Execute an HTTP get. • post = Execute an HTTP post.
namelist	list of identifiers	no	N/A	Space-separated list of variables submitted. If no namelist is specified, all named input item variables are submitted.
next	URI	no	N/A	URI for the HTTP request.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <help>, <if>, <noinput>, <nomatch>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form id="weather_info">
        <grammar src="../../grammar/weatherService.grxml" maxage="0"/>
        <block>
            <prompt bargein="false">
                Welcome to the weather information service.
                <audio src="http://www.online-ads.example/wis.wav"/>
                    Vox pilot develop anywhere deploy everywhere
            </prompt>
        </block>
        <initial name="start">
            <prompt>For what city and country would you like the weather?</prompt>
            <help>Please say the name of the city and country for which you would
                like a weather report. </help>
        </initial>
        <field name="country">
            <prompt>What country?</prompt>
            <help>Please speak the country for which you want the weather.</help>
        </field>
        <field name="city">
            <prompt>Please say the city in <value expr="country"/> for which you want
                the weather.</prompt>
            <help>Please speak the city for which you want the weather.</help>
        </field>
        <field name="go_ahead" type="boolean" modal="true">
            <prompt>
                Do you want to hear the weather for <value expr="city"/> in
                    <value expr="country"/>
            </prompt>
            <filled>
                <if cond="go_ahead">
                    <!-- Servlet example, does not exist -->
                        <submit next="/servlet/weather" method="post" namelist="city country"
                            fetchtimeout="45s" />
```

```
            </if>
            <clear namelist="start city country go ahead"/>
        </filled>
    </field>
  </form>
</vxml>

weatherService.grxml:
<?xml version="1.0"?>
<grammar version="1.0" root="weatherservice"
    xmlns="http://www.w3.org/2001/06/grammar"
    xml:lang="en-gb">
    <rule id="weatherservice">
        <one-of>
            <item>dublin <tag>dublin</tag></item>
            <item>cork <tag>cork</tag></item>
            <item>amsterdam <tag>amsterdam</tag></item>
        </one-of>
    </rule>
</grammar>
```

See Also

<goto>

<tag>

Contains the semantic interpretation of spoken user input.

Syntax

Attributes

None.

Details

The format of the <tag> element content depends on the tag-format attribute in the <grammar> element. For example, if the tag-format attribute is set to "semantics/1.0-literals", then literal text tokens are used. Typically, different ASR engines have different tag-formats. Refer to the ASR engine vendor documentation for more information.

Parents

None.

Children

None.

Example

```
<grammar xmlns="http://www.w3.org/2001/06/grammar"
    xml:lang="en-GB" root="ruleid" tag-format="semantics/1.0-literals">
    <rule id="ruleid" scope="public">
        <item>
            ten
            <tag>10/tag>
        </item>
    </rule>
</grammar>
```

<throw>

Raises a pre-defined or application-specific event.

Syntax

```
<throw
   event = "identifier"
   eventexpr = "CDATA"
   message = "CDATA"
   messageexpr = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
event	identifier	no	N/A	Pre-defined or application-specific event to raise. The raised event is available as the ECMAScript variable _event in the \<catch> element that handles the event.
eventexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the event to raise.
message	CDATA	no	N/A	String providing additional information about the event being raised. The message is available as the ECMAScript variable _message in the \<catch> element that handles the event.
messageexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the message.

Details

Use an event handler to catch the event. For information, see \<catch>.

Parents

\<block>, \<catch>, \<error>, \<filled>, \<foreach>, \<help>, \<if>, \<noinput>, \<nomatch>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <catch event="terminate">
        Caught terminate
        <goto next="#formExit"/>
    </catch>
    <help>
        <audio src="../../ProjectName/audio/help.wav" maxage="0"/>
            Help requested
        <goto next="#formEntry"/>
```

```
        </help>
        <error>
            Caught error
            <goto next="#formExit"/>
        </error>
        <var name="bCondition" expr="false"/>
        <form id="formEntry">
            <catch event="customEvent">
                Form scope custom handler
                Throwing terminate
                <throw event="terminate"/>
            </catch>
            <catch event="terminate" cond="bCondition">
                This handler will not be executed due to the condition
            </catch>
            <block>
                Throwing custom event
                <throw event="customEvent"/>
            </block>
        </form>
        <form id="formExit">
            <block>
                Goodbye
                <exit/>
            </block>
        </form>
</vxml>
```

<token>

Defines a word or phrases that can be spoken by the caller in an XML grammar.

Syntax

```
<token
    xml:lang = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
xml:lang	CDATA	no	N/A	Specifies the language used in the grammar token, conforming to RFC 1766.

Parents

<grammar>

Children

None.

Example

```
<?xml version= "1.0"?>
<grammar xmlns="http://www.w3.org/2001/06/grammar"
    xml:lang="en-GB" root="ruleid">
    <token>american express</token>
    <token>master card</token>
    <token>visa</token>
</grammar>
```

\<transfer\>

Transfers the call to the specified number.

Syntax

```
<transfer
   aai = "CDATA"
   aaiexpr = "CDATA"
   bridge = "{true | false}"
   cond = "CDATA"
   connecttimeout = "CDATA"
   dest = "URI"
   destexpr = "CDATA"
   expr = "CDATA"
   maxtime = "CDATA"
   name = "identifier"
   transferaudio = "URI"
   type = "{bridge | blind | consultation}"
    />
```

Attributes

Attribute	Data Type	Required?	Default	Description
aai	CDATA	no	N/A	String containing application-to-application (aai) data to send to an application at the far-end of the transfer.
aaiexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the aai.
bridge	boolean	no	no	Determines whether the platform stays in the connection with the caller and callee during and after the transfer. Valid values: - true = Document interpretation suspends until the transferred call terminates. - false = Raises the connection.disconnect.transfer event.
cond	CDATA	no	N/A	Boolean expression that must evaluate to ECMAScript true for the transfer to execute.
connecttimeout	CDATA	no	N/A	Time to wait for a connection to be made before the noanswer condition is returned.

Attribute	Data Type	Required?	Default	Description
dest	URI	no	N/A	URI of the transfer destination.
destexpr	CDATA	no	N/A	ECMAScript expression that evaluates to the URI of the transfer destination.
expr	CDATA	no	N/A	Initial value of the form item variable.
maxtime	CDATA	no	0	Maximum duration of the call if bridge is true.
name	CDATA	no	N/A	Stores the outcome of the transfer. Valid values: • busy = Destination refused the request. • noanswer = connecttimeout interval was exceeded before a connection occurred. • network_busy = Transfer was refused by an intermediary network. • near_end_disconnect = Transfer completed and was terminated by the caller. • far_end_disconnect = Transfer completed and was terminated by the callee. • network_disconnect = Transfer completed and was terminated by the network. • maxtime_disconnect = Transfer reached the maximum allowed duration and was terminated. • unknown = Transfer ended for an unknown reason. The form item shadow variable has the properties after transfer is completed (where *mycall* is the form item variable name):

Attribute	Data Type	Required?	Default	Description
				• *mycall*$.duration = Duration of a successful call, in seconds. • *mycall*$.utterance = User utterance, if the transfer was terminated by a recognition. • *mycall*$.inputmode = Input mode of the utterance (DTMF or voice), if the transfer was terminated by a recognition.
transferaudio	URI	no	N/A	URI of an audio file to play while attempting the transfer. By default, the audio used is a 2-tone (440 + 480 Hz), 2-second on, 4-second off sound. Only single-channel, 8 bit, 8 kHz audio files can be used for transferaudio.
type	NMTOKEN	no	blind	Type of transfer to perform. The <transfer> element can have a bridge attribute or a type attribute, but not both; otherwise an error.badfetch event is thrown. Valid values: • bridge = Equivalent to bridge is true. • blind = Equivalent to bridge is false. • consultation = Similar to a blind transfer, except that the disconnect event is only raised if the call was successfully transferred. Otherwise, the session between the original caller and the VoiceXML Interpreter remains active, and document execution resumes.

Details

The following events can be raised by a transfer:

Event	Description
error.connection.baddestination	Destination URI is malformed.
connection.disconnect.hangup	Caller hung up.
connection.disconnect.transfer	Call was transferred (see the bridge attribute).
error.connection.noauthorization	Caller is not allowed to call the destination.
error.connection.noresource	Platform cannot allocate resources to place the call.
error.connection.noroute	Not implemented.
error.connection.protocol.nnn	Not implemented.
error.unsupported.uri	URI format used in the destination number is not supported.

Parents

<form>

Children

<audio>, <catch>, <enumerate>, <error>, <filled>, <grammar>, <help>, <noinput>, <nomatch>, <prompt>, <property>, <value>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <var name="phoneNumber" expr="'tel:+35312345678'"/>
        <block>Transferring your call, please hold</block>
        <transfer destexpr="phoneNumber" name="callTransfer" type="bridge"
            connecttimeout="10s">
            <filled>
                Your transfer lasted
                <value expr="callTransfer$.duration" /> seconds.
                <if cond="callTransfer == 'busy
                    Your call party is busy please call back later
                </if>
            </filled>
        </transfer>
    </form>
</vxml>
```

See Also

<form>

<value>

Returns the value of a variable.

Syntax

```
<value
   expr = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
expr	CDATA	no	application/x-www-form-urlencoded	ECMAScript expression that evaluates to the variable's value.

Parents

<audio>, <block>, <catch>, <emphasis>, <enumerate>, <error>, <field>, <filled>, <foreach>, <help>, <if>, <initial>, <log>, <menu>, <noinput>, <nomatch>, <object>, <p>, <prompt>, <prosody>, <record>, <s>, <say-as>, <subdialog>, <transfer>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
   <var name="number"/>
    <var name="month" expr="'March'"/>
    <form>
       <block>
           Month is <value expr="month"/>
           <assign name="month" expr="'July'"/>
           Now month is <value expr="month"/>
           <assign name="number" expr="2*5"/>
           Number is <value expr="number"/>
       </block>
    </form>
</vxml>
```

<var>

Declares a local variable.

Syntax

```
<var
   name = "identifier"
   expr = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
name	identifier	yes	N/A	Variable name. This must be a legal ECMAScript identifier.

Attribute	Data Type	Required?	Default	Description
expr	CDATA	no	ECMAScript undefined	Initial value of the variable.

Parents

<block>, <catch>, <error>, <filled>, <foreach>, <form>, <help>, <if>, <noinput>, <nomatch>, <vxml>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <var name="number"/>
    <var name="month" expr="'March'"/>
    <form>
        <block>
            Month is <value expr="month"/>
            <assign name="month" expr="'July'"/>
            Now month is <value expr="month"/>
            <assign name="number" expr="2*5"/>
            Number is <value expr="number"/>
        </block>
    </form>
</vxml>
```

See Also

<assign>, <script>

<voice>

Voice characteristics for the spoken text.

Syntax

```
<voice
   age = "CDATA"
   gender = "{male | female | neutral}"
   name = "CDATA"
   xml:lang = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
age	CDATA	no	N/A	Integer specifying the preferred age of the voice to speak the contained text.
gender	{male | female | neutral}	no	N/A	Preferred gender of the voice to speak the contained text.

Attribute	Data Type	Required?	Default	Description
name	CDATA	no	N/A	Name of the voice to speak the contained text.
xml:lang	CDATA	no	N/A	RFC 1766 compliant identifier used to locate the TTS engine used to render the enclosed text.

Parents

<audio>, <emphasis>, <enumerate>, <p>, <prompt>, <prosody>, <s>, <voice>

Children

<audio>, <break>, <emphasis>, <mark>, <p>, <phoneme>, <prosody>, <s>, <say-as>, <sub>, <value>, <voice>

Example

```
<?xml version="1.0"?>
 <vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <prompt>
                <voice gender="female" age="5">Mary had a little lamb,</voice>
                <!-- now request a different female child's voice -->
                <voice gender="female" age="6" xml:lang="en-GB">
                    It's fleece was white as snow.
                </voice>
                <!-- platform-specific voice selection -->
                <voice name="Mike">I want to be like Mike.</voice>
            </prompt>
        </block>
    </form>
</vxml>
```

See Also

<emphasis>, <prosody>, <say-as>

<vxml>

Top-level element required in every VoiceXML document.

Syntax

```
<vxml
   application = "URI"
   version = "CDATA"
   xml:base = "URI"
   xml:lang = "CDATA"
   xmlns = "CDATA"
   xmlns:xsi = "CDATA"
   xsi:schemalocation = "CDATA"
/>
```

Attributes

Attribute	Data Type	Required?	Default	Description
application	URI	no	N/A	URI of this document's application root document.
version	CDATA	yes	N/A	VoiceXML version.
xml:base	URI	no	N/A	Base URI to use when resolving relative URIs in the document.
xmlns:dlgc				
xml:lang	CDATA	no	N/A	Language variant of the document.
xmlns	CDATA	no	N/A	Designated namespace for VoiceXML.
xmlns:xsi	CDATA	no	N/A	Used with the xsi:schemalocation attribute to indicate the location of the schema for the VoiceXML namespace.
xsi:schemalocation	CDATA	no	N/A	Used with the xmlns:xsi attribute to indicate the location of the schema for the VoiceXML namespace.

Parents

None.

Children

<catch>, <data>, <error>, <form>, <help>, <link>, <menu>, <meta>, <metadata>, <noinput>, <nomatch>, <property>, <script>, <var>

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            Hello World!
        </block>
    </form>
</vxml>
```

5. Appendix: VoiceXML Elements and Attributes

The following tables list the VoiceXML elements and attributes available in PowerMedia XMS.

- VXML 2.0
- VXML 2.1
- SRGS 1.0

Clicking on the section number in the **Attributes** column will bring you to the W3C document for a detailed description of the element and its attributes. The **XMS 3.3** column shows if it is supported in PowerMedia XMS Release 3.3.

X = Supported, N = Not Supported

VXML 2.0

Element	Attributes	Description	XMS 3.3
<assign>	5.3.2	Assign a variable a value	X
	Name	The name of the variable being assigned to	X
	expr	The new value of the variable	X
<audio>	4.1.3	Play an audio clip within a prompt	X
	src	The URI of the audio prompt	X
	fetchhint	Specifies when the VoiceXML Interpreter context should retrieve content from the server	N
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	N
	maxage	Age is no greater than the specified time in second	N
	maxstale	Age expiration delay supported	N
	expr	ECMAScript expression	X
<block>	2.3.2	A container of (non-interactive) executable code	X

Element	Attributes	Description	XMS 3.3
	name	The name of the form item	X
	expr	The initial value of the form item	X
	cond	Condition in order for the form item to be visited	X
<catch>	5.2.2	Catch an event	X
	event	The event or events to catch	X
	count	Counter of the event	X
	cond	Condition of the event	X
<choice>	2.2.2	Define a menu item	X
	dtmf	The DTMF sequence for this choice	X
	accept	Override the setting for accept in< menu> for this particular choice	X
	next	The URI of next dialog or document	X
	expr	Specify an expression to evaluate as a URI to transition to instead of specifying a next	X
	event	Specify an event to be thrown instead of specifying a next	X
	eventexpr	An ECMAScript expression evaluating to the name of the event to be thrown	X
	message	A message string providing additional context about the event being thrown	X
	messageexpr	An ECMAScript expression evaluating to the message string	X
	fetchaudio	The URI of the audio clip to play while the fetch is being done	X

Element	Attributes	Description	XMS 3.3
	fetchhint	Defines when the interpreter context should retrieve content from the server	X
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	X
	maxage	Age is no greater than the specified time in second	X
	maxstale	Age expiration delay supported	X
<clear>	5.3.3	Clear one or more form item variables	X
	namelist	The list of variables to be reset	X
<disconnect>	5.3.11	Disconnect a session	X
	None		
<else>	5.3.4	Used in <if> elements	X
	None		
<elseif>	5.3.4	Used in <if> elements	X
	cond	Boolean expression that must evaluate to ECMAScript true for the <elseif> element to execute	X
<enumerate>	2.2.4	Shorthand for enumerating the choices in a menu	X
	None		

Element	Attributes	Description	XMS 3.3
<error>	5.2.3	Catch an error event	X
	count	The event count	X
	cond	An optional condition to test to see if the event is caught by this element	X
<exit>	5.3.9	Exit a session	X
	expr	An ECMAScript expression that is evaluated as the return value	X
	namelist	Variable names to be returned to interpreter context	X
<field>	2.3.1	Declares an input field in a form	X
	name	The form item variable in the dialog scope that will hold the result	X
	expr	The initial value of the form item variable	X
	cond	An expression that must evaluate to true after conversion to boolean in order for the form item to be visited	X
	type	The type of field	X
	slot	The name of the grammar slot used to populate the variable	X
	modal	If this is false (the default), all active grammars are turned on while collecting this field. If this is true, then only the field's grammars are enabled. All others are temporarily disabled.	X
<filled>	2.4	An action executed when fields are filled	X

Element	Attributes	Description	XMS 3.3
	mode	Either all (the default), or any	X
	namelist	The input items to trigger on	X
<form>	2.1	A dialog for presenting information and collecting data	X
	id	The name of the form	X
	scope	The default scope of the form's grammars (dialog or document)	X
<goto>	5.3.7	Go to another dialog in the same or different document	X
	next	The URI to which to transition	X
	expr	An ECMAScript expression that yields the URI	X
	nextitem	The name of the next form item to visit in the current form	X
	expritem	An ECMAScript expression that yields the name of the next form item to visit	X
	fetchaudio	The URI of the audio clip to play while the fetch is being done	X
	fetchhint	Defines when the interpreter context should retrieve content from the server	X
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	X
	maxage	Age is no greater than the specified time in second	X
	maxstale	Age expiration delay supported	X

Element	Attributes	Description	XMS 3.3
<grammar>	3.1	Specify a speech recognition or DTMF grammar	X
	version	Defines the version of the grammar	X
	xml:lang	The language identifier of the grammar	X
	xmlns	Designated namespace for the SRGS XML format Stand-alone SRGS XML documents only	X
	xmlns:xsi	Used with the xsi:schemalocation attribute to indicate the location of the schema for the SRGS XML namespace Stand-alone SRGS XML documents only	X
	xsi:schemalocation	Used with xmlns:xsi attribute to indicate the location of the schema for the SRGS XML namespace	X
	mode	Defines the mode of the grammar	X
	root	Defines the rule which acts as the root rule of the grammar	X
	tag-format	Defines the tag content format for all tags within the grammar	X
	xml:base	Declares the base URI from which relative URIs in the grammar are resolved	X
	src	The URI specifying the location of the grammar and optionally a rulename within that grammar, if it is external	X
	scope	Either document, which makes the grammar active in all dialogs of the current document (and relevant application leaf documents), or dialog, to make the grammar active throughout the current form	X

Element	Attributes	Description	XMS 3.3
	type	The preferred media type of the grammar	X
	weight	Specifies the weight of the grammar	X
	fetchhint	Defines when the interpreter context should retrieve content from the server	X
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	X
	maxage	Age is no greater than the specified time in second	X
	maxstale	Age expiration delay supported	X
<help>	5.2.3	Catch a help event	X
	count	The event count (as in <catch>)	X
	cond	An optional condition to test to see if the event is caught by this element	X
<if>	5.3.4	Simple conditional logic	X
	cond	Boolean value of TRUE or FALSE	X
<initial>	2.3.3	Declares initial logic upon entry into a (mixed initiative) form	X
	name	The name of a form item variable used to track whether the <initial> is eligible to execute	X
	expr	The initial value of the form item variable; default is ECMAScript undefined. If initialized to a value, then the form item will not be visited unless the form item variable is cleared.	X

Element	Attributes	Description	XMS 3.3
	cond	An expression that must evaluate to true after conversion to boolean in order for the form item to be visited	X
<link>	2.5	Specify a transition common to all dialogs in the link's scope	X
	next	The URI to go to	X
	expr	Like next, except that the URI is dynamically determined by evaluating the given ECMAScript expression	X
	event	The event to throw when the user matches one of the link grammars	X
	eventexpr	An ECMAScript expression evaluating to the name of the event to throw when the user matches one of the link grammars	X
	message	A message string providing additional context about the event being thrown	N
	messageexpr	An ECMAScript expression evaluating to the message string	N
	dtmf	The DTMF sequence for this link	X
	fetchaudio	The URI of the audio clip to play while the fetch is being done	X
	fetchhint	Defines when the interpreter context should retrieve content from the server	X
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	X
	maxage	Age is no greater than the specified time in second	N
	maxstale	Age expiration delay supported	N

Element	Attributes	Description	XMS 3.3
<log>	5.3.13	Generate a debug message	X
	label	An optional string which may be used (e.g., to indicate the purpose of the log)	X
	expr	An optional ECMAScript expression evaluating to a string	X
<mark>	3.3.2	The mark element can be used to reference a specific location in the text/tag sequence, and can additionally be used to insert a marker into an output stream for asynchronous notification	X
	name	Attribute identifying to inform the hosting environment the reference when the corresponding position has been reached	X
<menu>	2.2.1	A dialog for choosing amongst alternative destinations	X
	id	The identifier of the menu	X
	scope	The menu's grammar scope. Either dialog (the default) or document.	X
	dtmf	When set to true, the first nine choices that have not explicitly specified a value for the dtmf attribute are given the implicit ones 1, 2, etc. Remaining choices that have not explicitly specified a value for the dtmf attribute will not be assigned DTMF values (and thus cannot be matched via a DTMF keypress). If there are choices which have specified their own DTMF sequences to be something other than *, #, or 0, an error.badfetch will be thrown. The default is false.	X

Element	Attributes	Description	XMS 3.3
	accept	When set to exact (the default), the text of the choice elements in the menu defines the exact phrase to be recognized. When set to approximate, the text of the choice elements defines an approximate recognition phrase (as described under 2.2.5). Each <choice> can override this setting.	X
<meta>	6.2.1	Define a metadata item as a name/value pair	X
	name	The name of the metadata property	X
	content	The value of the metadata property	X
	http-equiv	The name of an HTTP response header	X
<metadata>	6.2.2	Define metadata information using a metadata schema	X
	Creator	An entity primarily responsible for making the content of the resource	N
	Rights	Information about rights held in and over the resource	N
	Subject	The topic of the content of the resource	N
<noinput>	5.2.3	Catch a noinput event	X
	count	The event count (as in <catch>)	X
	cond	An optional condition to test to see if the event is caught by this element	X
<nomatch>	5.2.3	Catch a nomatch event	X

Element	Attributes	Description	XMS 3.3
	count	The event count (as in <catch>)	X
	cond	An optional condition to test to see if the event is caught by this element	X
<object>	2.3.5	Interact with a custom extension	X
	name	When the object is evaluated, it sets this variable to an ECMAScript value whose type is defined by the object	X
	expr	The initial value of the form item variable; default is ECMAScript undefined. If initialized to a value, then the form item will not be visited unless the form item variable is cleared.	X
	cond	An expression that must evaluate to true after conversion to boolean in order for the form item to be visited	X
	classid	The URI specifying the location of the object's implementation.	X
	codebase	The base path used to resolve relative URIs specified by classid, data, and archive. It defaults to the base URI of the current document.	N
	codetype	The content type of data expected when downloading the object specified by classid. When absent it defaults to the value of the type attribute.	N
	data	The URI specifying the location of the object's data. If it is a relative URI, it is interpreted relative to the codebase attribute.	N
	type	The content type of the data specified by the data attribute	N

Element	Attributes	Description	XMS 3.3
	archive	A space-separated list of URIs for archives containing resources relevant to the object, which may include the resources specified by the classid and data attributes. URIs which are relative are interpreted relative to the codebase attribute.	N
	fetchhint	Defines when the interpreter context should retrieve content from the server	N
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	N
	maxage	Age is no greater than the specified time in second	N
	maxstale	Age expiration delay supported	N
<option>	2.3.1.3	Specify an option in a <field>	X
	dtmf	An optional DTMF sequence for this option	X
	accept	When set to exact (the default), the text of the option element defines the exact phrase to be recognized. When set to approximate, the text of the option element defines an approximate recognition phrase.	X
	value	The string to assign to the field's form item variable when a user selects this option, whether by speech or DTMF.	X
<param>	6.4	Parameter in <object> or <subdialog>	X
	name	The name to be associated with this parameter when the object or subdialog is invoked	X

Element	Attributes	Description	XMS 3.3
	expr	An expression that computes the value associated with name	X
	value	Associates a literal string value with name	X
	valuetype	One of data or ref, by default data; used to indicate to an object if the value associated with name is data or a URI (ref)	N
	type	The media type of the result provided by a URI if the valuetype is ref; only relevant for uses of <param> in <object>	N
<prompt>	4.1	Queue speech synthesis and audio output to the user	X
	bargein	Control whether a user can interrupt a prompt	X
	bargeintype	Sets the type of bargein to be speech, or hotword	X
	cond	An expression that must evaluate to true after conversion to boolean in order for the prompt to be played. Default is true.	X
	count	A number that allows you to emit different prompts if the user is doing something repeatedly. If omitted, it defaults to 1.	X
	timeout	The timeout that will be used for the following user input. The value is a time designation.	X
	xml:lang	The language identifier for the prompt. If omitted, it defaults to the value specified in the document's xml:lang attribute.	X

Element	Attributes	Description	XMS 3.3
	xml:base	Declares the base URI from which relative URIs in the prompt are resolved	N
<property>	6.3	Control implementation platform settings. See 6.3 for a list of generic properties defined by VoiceXML.	X
	name	The name of the property	X
	value	The value of the property	X
<record>	2.3.6	Record an audio sample	X
	name	The input item variable that will hold the recording	X
	expr	The initial value of the form item variable; default is ECMAScript undefined. If initialized to a value, then the form item will not be visited unless the form item variable is cleared.	X
	cond	An expression that must evaluate to true after conversion to boolean in order for the form item to be visited	X
	modal	If this is true (the default) all non-local speech and DTMF grammars are not active while making the recording. If this is false, non-local speech and DTMF grammars are active.	X
	beep	If true, a tone is emitted just prior to recording. Defaults to false.	X
	maxtime	The maximum duration to record	X
	finalsilence	The interval of silence that indicates end of speech	N

Element	Attributes	Description	XMS 3.3
	dtmfterm	If true, any DTMF keypress not matched by an active grammar will be treated as a match of an active (anonymous) local DTMF grammar. Defaults to true.	X
	type	The media format of the resulting recording	X
<reprompt>	5.3.6	Play a field prompt when a field is re-visited after an event	X
	None		
<return>	5.3.10	Return from a subdialog	X
	event	Return, then throw this event	X
	eventexpr	Return, then throw the event to which this ECMAScript expression evaluates	X
	message	A message string providing additional context about the event being thrown	X
	messageexpr	An ECMAScript expression evaluating to the message string	X
	namelist	Variable names to be returned to calling dialog	X
<script>	5.3.12	Specify a block of ECMAScript client-side scripting logic	X
	src	The URI specifying the location of the script, if it is external	X
	charset	The character encoding of the script designated by src	X

Element	Attributes	Description	XMS 3.3
	fetchhint	Defines when the interpreter context should retrieve content from the server	X
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	X
	maxage	Age is no greater than the specified time in second	X
	maxstale	Age expiration delay supported	X
<subdialog>	2.3.4	Invoke another dialog as a subdialog of the current one	X
	name	The result returned from the subdialog, an ECMAScript object whose properties are the ones defined in the namelist attribute of the <return> element	X
	expr	The initial value of the form item variable	X
	cond	An expression that must evaluate to true after conversion to boolean in order for the form item to be visited	X
	namelist	The list of variables to submit. The default is to submit no variables.	X
	src	The URI of the subdialog	X
	srcexpr	An ECMAScript expression yielding the URI of the subdialog	X
	method	The request method: get (the default) or post	X
	enctype	The media encoding type of the submitted document (when the value of method is "post")	X
	fetchaudio	The URI of the audio clip to play while the fetch is being done	X

Element	Attributes	Description	XMS 3.3
	fetchtimeout	Defines when the interpreter context should retrieve content from the server	X
	fetchhint	The interval to wait for the content to be returned before throwing an error	X
	maxage	Age is no greater than the specified time in second	X
	maxstale	Age expiration delay supported	X
<submit>	5.3.8	Submit values to a document server	X
	next	The URI reference	X
	expr	Like next, except that the URI reference is dynamically determined by evaluating the given ECMAScript expression	X
	namelist	The list of variables to submit. By default, all the named input item variables are submitted	X
	method	The request method: get (the default) or post	X
	enctype	The media encoding type of the submitted document (when the value of method is post)	X
	fetchaudio	The URI of the audio clip to play while the fetch is being done	X
	fetchhint	Defines when the interpreter context should retrieve content from the server	X
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	X
	maxage	Age is no greater than the specified time in second	X
	maxstale	Age expiration delay supported	X

Element	Attributes	Description	XMS 3.3
<throw>	5.2.1	Throw an event	X
	event	The event being thrown	X
	eventexpr	An ECMAScript expression evaluating to the name of the event being thrown	X
	message	A message string providing additional context about the event being thrown	X
	messageexpr	An ECMAScript expression evaluating to the message string	X
<transfer>	2.3.7	Transfer the caller to another destination	X
	name	Stores the outcome of a bridge transfer attempt. In the case of a blind transfer, this variable is undefined.	X
	expr	The initial value of the form item variable; default is ECMAScript undefined	X
	cond	An expression that must evaluate to true in order for the form item to be visited	X
	dest	The URI of the destination (telephone, IP telephony address)	X
	destexpr	An ECMAScript expression yielding the URI of the destination	X

Element	Attributes	Description	XMS 3.3
	bridge	Determines whether the platform remains in the connection with the caller and callee. bridge=true *Bridge transfer*. The platform adds the callee to the connection. Document interpretation suspends until the transferred call terminates. The platform remains in the connection for the duration of the transferred call; listening during transfer is controlled by any included< grammar>s. If the caller disconnects by going onhook or if the network disconnects the caller, the platform throws a connection.disconnect.hangup event. If the connection is released for any other reason, that outcome is reported in the name attribute (see the following table). bridge=false *Blind transfer* (default). The platform redirects the caller to the callee without remaining in the connection, and does not monitor the outcome. The platform throws a connection.disconnect.transfer immediately, regardless of whether the transfer was successful or not.	X
	connecttimeout	The time to wait while trying to connect the call before returning the noanswer condition	X
	maxtime	The time that the call is allowed to last, or 0s if no limit is imposed	X
	transferaudio	The URI of audio source to play while the transfer attempt is in progress (before far-end answer)	X
	aai	Application-to-application information. A string containing data sent to an application on the far-end, available in the session variable session.connection.aai.	X

Element	Attributes	Description	XMS 3.3
	aaiexpr	An ECMAScript expression yielding the AAI data	X
<value>	4.1.4	Insert the value of an expression in a prompt	X
	expr	The expression to render	X
<var>	5.3.1	Declare a variable	X
	name	The name of the variable that will hold the result	X
	expr	The initial value of the variable (optional). If there is no expr attribute, the variable retains its current value, if any.	X
<vxml>	1.5.1	Top-level element in each VXML document	X
	version	The version of VXML of this document (required). The current version number is 2.0.	X
	xmlns	The designated namespace for VXML (required). The namespace for VXML is defined to be http://www.w3.org/2001/vxml.	X
	xml:base	The base URI for this document as defined in [XML-BASE]	X
	xml:lang	The language identifier for this document. If omitted, the value is a platform-specific default.	X
	application	The URI of this document's application root document, if any	X

Element	Attributes	Description	XMS 3.3
	Xmlns:xsi	Used with the xsi:schemalocation attribute to indicate the location of the schema for the VXML namespace	X
	Xsi:schemalocation	Used with the xmlns:xsi attribute to indicate the location of the schema for the VXML namespace	X

VXML 2.1

Element	Attributes	Description	XMS 3.3
<data>	5	Fetches arbitrary XML data from a document server	X
	src	The URI specifying the location of the XML data to retrieve	X
	name	The name of the variable that exposes the DOM	X
	srcexpr	Like src, except that the URI is dynamically determined by evaluating the given ECMAScript expression when the data needs to be fetched	X
	method	The request method: get (the default) or post	X
	namelist	The list of variables to submit	X
	enctype	The media encoding type of the submitted document	X
	fetchaudio	The URI of the audio clip to play while the fetch is being done	X
	fetchhint	Defines when the interpreter context should retrieve content from the server	X
	fetchtimeout	The interval to wait for the content to be returned before throwing an error	X
	maxage	Age is no greater than the specified time in second	X

Element	Attributes	Description	XMS 3.3
	maxstale	Age expiration delay supported	X
<disconnect>	8	Disconnects a session (VXML 2.1 extends the <disconnect> element to support the following attribute)	
	namelist	Variable names to be returned to the interpreter context	X
<grammar>	2	References a speech recognition or DTMF grammar (VXML 2.1 extends the <grammar> element to support the following additional attribute)	
	srcexpr	Equivalent to src, except that the URI is dynamically determined by evaluating the given ECMAScript expression in the current scope (the current form item)	X
<foreach>	6	Iterates through an ECMAScript array	X
	array	An ECMAScript expression that must evaluate to an ECMAScript array	X
	item	The variable that stores each array item upon each iteration of the loop	X
<mark>	4	Declares a bookmark in a sequence of prompts (VXML 2.1 extends the <mark> element to support the following additional attribute)	X
	nameexpr	An ECMAScript expression which evaluates to the name of the mark when the prompt is queued	X

Element	Attributes	Description	XMS 3.3
<property>	5.1, 7	Controls platform settings (New properties to <data>, <field>, <initial>, <menu>, <record> and <transfer>)	X
	name		X
	value		X
<script>	3	References a document containing client-side ECMAScript (VXML 2.1 extends the <script> element to support the following additional attribute)	X
	srcexpr	Equivalent to src, except that the URI is dynamically determined by evaluating the given ECMAScript expression	X
<transfer>	9	Transfers the user to another destination (VXML 2.1 extends the <transfer> element to support the following additional attribute)	X
	type	The type of transfer. The value can be bridge, blind, or consultation.	X
SSML 1.0 for VXML			
<audio>	3.3.1	Specifies audio files to be played and text to be spoken	X
		Refer to VXML 2.0 <audio> element	
<break>	3.2.3	Specifies a pause in the speech output	X

Element	Attributes	Description	XMS 3.3
	strength	Optional attribute having one of the following values: none, x-weak, weak, medium (default value), strong, or x-strong	X
	time	Optional attribute indicating the duration of a pause to be inserted in the output in seconds or milliseconds	X
<desc>	3.3.3	Provides a description of a non-speech audio source in <audio>	X
	xml:lang	Indicates that the content of the element is in a different language from that of the content surrounding the element	X
<emphasis>	3.2.2	Specifies that the enclosed text should be spoken with emphasis	X
	level	Indicates the strength of emphasis to be applied. Defined values are strong, moderate, none, or reduced.	X
<lexicon>	3.1.4	Specifies a pronunciation lexicon for the prompt	X
	type	Specifies the media type of the pronunciation lexicon document	X
	uri	Specifies a URI that identifies the location of the pronunciation lexicon document	X
<meta>	3.1.5	Containers in which information about the document can be placed	X
	name	Property name of the meta. seeAlso is the only defined property name supported.	X

Element	Attributes	Description	XMS 3.3
	http-equiv	Used when documents are retrieved via HTTP	X
	content	Information about the content of the document	X
<metadata>	3.1.6	Specifies XML metadata content for the prompt	X
	None		
<p>	3.1.7	Identifies the enclosed text as a paragraph, containing zero or more sentences	X
	xml:lang	Indicates the natural language of the enclosing element and its attributes and subelements	X
<phoneme>	3.1.9	Specifies a phonetic pronunciation for the contained text	X
	ph	Specifies the phoneme/phone string	X
	alphabet	Optional attribute that specifies the phonemic/phonetic alphabet	X
<prosody>	3.2.4	Specifies prosodic information for the enclosed text. Control the pitch, speaking rate and volume of the speech output.	X
	pitch	The baseline pitch for the contained text	X
	contour	Sets the actual pitch contour for the contained text	X

Element	Attributes	Description	XMS 3.3
	range	The pitch range (variability) for the contained text	X
	rate	A change in the speaking rate for the contained text	X
	duration	A value in seconds or milliseconds for the desired time to take to read the element contents	X
	volume	The volume for the contained text in the range 0.0 to 100.0	X
<say-as>	3.1.8	Specifies the type of text construct contained within the element	X
	interpret-as	Indicates the content type of the contained text construct	X
	format	Gives further hints on the precise formatting of the contained text for content types that may have ambiguous formats	X
	detail	Indicates the level of detail to be read aloud or rendered	X
<s>	3.1.7	Identifies the enclosed text as a sentence	X
	xml:lang	Indicates the natural language of the enclosing element and its attributes and subelements	X
<speak>	3.1.1	Top-level element required in a standalone SSML document	X
	version	VXML version	X
	xml:base	Base URI, used to resolve relative URIs in the document	X

Element	Attributes	Description	XMS 3.3
	xml:lang	Language variant of the document	X
	xmlns	Designated namespace for VXML	X
	xmlns:xsi	Used with the xsi:schemalocation attribute to indicate the location of the schema for the VXML namespace	X
	xsi:schemalocation	Used with the xmlns:xsi attribute to indicate the location of the schema for the VXML namespace	X
<sub>	3.1.10	Specifies replacement spoken text for the contained text	X
	alias	Specifies the string to be spoken instead of the enclosed string	X
<voice>	3.2.1	Specifies voice characteristics for the spoken text	X
	xml:lang	Indicates the natural language of the enclosing element and its attributes and subelements	X
	gender	Indicates the preferred gender of the voice to speak the contained text	X
	age	Indicates the preferred age in years (since birth) of the voice to speak the contained text	X
	variant	Indicates a preferred variant of the other voice characteristics to speak the contained text	
	name	Indicates a processor-specific voice name to speak the contained text	X

SRGS 1.0

Element	Attributes	Description	XMS 3.3
<grammar>	4	Root element of an XML grammar	X
		Refer to VXML 2.0 <grammar> and VXML 2.1 <grammar> for a list of supported attributes	
<meta>	4.11.1	Header declaration of meta content of an HTTP equivalent	X
	name	Property name of the meta. seeAlso is the only defined property name supported.	X
	http-equiv	Used when documents are retrieved via HTTP	X
	content	Information about the content of the document	X
<metadata>	4.11.2	Header declaration of XML metadata content	X
	None		
<lexicon>	4.10	Header declaration of a pronunciation lexicon	X
	type	Specifies the media type of the pronunciation lexicon document	X
	uri	Specifies a URI that identifies the location of the pronunciation lexicon document	X
<rule>	3	Declare a named rule expansion of a grammar	X

Element	Attributes	Description	XMS 3.3
	id	Indicates the name of the rule and must be unique within the grammar	X
	scope	Defines the scope of the rule definition. Either public or private.	X
<token>	2.1	Define a word or other entity that may serve as input	X
	xml-lang	Indicates the language of the contained token	X
<ruleref>	2.2	Refer to a rule defined locally or externally	X
	uri	Defines the URI of the referenced grammar and rule within the ruleref	X
	type	Specifies the media type of the grammar containing the reference	X
<item>	2.3	Define an expansion with optional repeating and probability	X
	weight	A weight is nominally a multiplying factor in the likelihood domain of a speech recognition search	X
	repeat	Indicates the number of times the contained expansion may be repeated	X
	repeat-prob	Carries the repeat probability. Repeat probabilities are supported on any item element but are ignored if the repeat attribute is not also specified.	X
<one-of>	2.4	Defines a set of alternative rule expansions	X

Element	Attributes	Description	XMS 3.3
	xml-lang	Indicates the language of the contained alternatives	X
<example>	3.3	Element contained within a rule definition that provides an example of input that matches the rule	X
	None		
<tag>	2.6	Defines an arbitrary string that to be included inline in an expansion which may be used for semantic interpretation	X
	None		