



VoiceGenie VoiceXML Gateway

Developer Guide

1120 Finch Avenue W., 8th Floor, Toronto, ON Canada M3J 3H7 T-416-736-0905 F-416-736-1551 www.voicegenie.com

Version: 5.0

Release Date: October 23, 2001

Table of Contents

VoiceXML 2.0 Reference	4
Event Handling	5
Variables Reference	7
Properties Reference	9
<assign>.....	14
<audio>	15
<block>	17
<catch>.....	18
<choice>.....	20
<clear>.....	22
<disconnect>.....	24
<else>.....	25
<elseif>.....	26
<enumerate>.....	27
<error>.....	29
<exit>.....	31
<field>.....	32
<filled>.....	36
<form>.....	38
<goto>.....	40
<grammar>	43
<help>.....	46
<if>.....	47
<initial>	48
<link>.....	50
<log>.....	52
<menu>	54
<meta>.....	56
<noinput>.....	57
<nomatch>.....	58
<object>	59
<option>.....	61
<param>	63
<prompt>	65
<property>.....	67
<record>.....	69
<reprompt>.....	72
<return>.....	73
<script>	75
<subdialog>.....	77
<submit>	79
<throw>	81
<transfer>	83
<value>.....	86
<var>.....	87
<vxml>	88
Speech Markup Reference	90
<break>	90
<emphasis>.....	91
<paragraph>.....	91
<phoneme>.....	92
<prosody>.....	92

<say-as>.....	93
<sentence>.....	94
<voice>.....	95
ECMAScript Reference	97

VoiceXML 2.0 Reference

Voice Extensible Markup Language (VoiceXML) is a markup language designed for telephony-based applications that features synthesized speech, speech recognition, and DTMF inputs. This document provides a reference to VoiceGenie's VoiceXML implementation, which is compliant to the W3C VoiceXML 2.0 specification.

Tutorials

VoiceXML tutorials are available online on VoiceGenie's SpeechGenie Workshop. You will find step-by-step instructions on how to get started with VoiceXML quickly.

<http://www.speechgenie.com/tutorials.php>

VoiceGenie Features

VoiceGenie extends VoiceXML by adding the following features:

Call Control Extensions

<http://www.speechgenie.com/tutorials.php?tutorial=callcontrol>

Audio Control

http://www.speechgenie.com/tutorials.php?tutorial=audio_control

Audio Offset - manipulation of audio playback

http://www.speechgenie.com/tutorials.php?tutorial=audio_offset_sg

Saving Utterances

http://www.speechgenie.com/tutorials.php?tutorial=saved_utterance_sg

Reference

Please refer to the following links for additional references:

W3C VoiceXML 2.0 Specification

<http://www.w3.org/TR/2001/WD-voicexml20-20011023/>

VoiceXML 2.0 Tag Reference (this document)

<http://www.speechgenie.com/vxml2ref.php>

VoiceXML 2.0 Tag Summary

<http://www.speechgenie.com/vxml2ref.php?ref=summary>

Grammar Reference

http://www.speechgenie.com/tutorials.php?tutorial=osr_sg

Speech Markup Reference

<http://www.speechgenie.com/vxml2ref.php?ref=SSML>

ECMAScript Quick Reference

<http://www.speechgenie.com/vxml2ref.php?ref=ECMAScript>

OpenSpeech DialogModules

<http://www.speechgenie.com/osdminfo.php>

Event Handling

The VoiceXML interpreter throws events for many different situations: application errors, user does not respond, etc. VoiceXML defines a set of elements to catch common events thrown by the interpreter: <catch>, <error>, <help>, <noinput>, and <nomatch>. An element inherits the catch elements from each of its ancestor elements, as needed. For example, if a <field> element does not contain a <nomatch > element, but its parent <form> does, the form's nomatch element is used. In this way, common event handling behavior can be specified at any level, and it applies to all descendents.

Events

VoiceGenie has defined additional events to support Call Control Extensions. Please refer to the [Call Control Extension Tutorial](#) for detailed information.

Event	Description
cancel	The user has requested to cancel playing of the current prompt. (Available when the universals property includes cancel)
exit	The user has asked to exit. (Available when the universals property includes exit)
help	The user has asked for help. (Available when the universals property includes help)
noinput	The user has not responded within the timeout interval.
nomatch	The user input something, but it was not recognized.
maxspeectimeout	The user input was too long, exceeding the property maxspeectimeout.
telephone.disconnect.hangup	Caller hangs up.
telephone.disconnect.transfer	Call was "blind transferred" to another line and will not return.

Errors

Error	Description
error.badfetch	The interpreter context throws this event when a fetch of a resource has failed <i>and</i> the interpreter context has reached a place in the document interpretation where the fetch result is required.
error.badfetch.http.<response code>	In the case of a fetch failure, the interpreter context returns the specific HTTP response code.
error.semantic	A run-time error was found in the VoiceXML document.
error.noauthorization	The user is not authorized to perform the requested operation.
error.unsupported.format	The requested resource has a format that is not supported by the platform.
error.unsupported.language	The platform does not support the language for either speech synthesis or speech recognition.
error.telephone.noauthorization	Caller is not allowed to call the destination.
error.telephone.baddestination	Destination URI is malformed.
error.telephone.noroute	The platform is not able to place a call to the destination.
error.telephone.noresource	The platform cannot allocate resource to place the call.
error.asr (VoiceGenie Extension)	ASR error.
error.grammar.dtmf (VoiceGenieExtension)	There is an error in the DTMF grammar.
error.application (VoiceGenieExtension)	Thrown when there is a miscellaneous application error.
error.script (VoiceGenie Extension)	There is an error with ECMAScript.
error.internal (VoiceGenie Extension)	Internal VoiceGenie server error.

Default Catch Elements

The interpreter provides the following events with a default event handler.

Event	Audio Provided	Action
cancel	no	don't reprompt
error	yes	exit interpreter
exit	no	exit interpreter
help	yes	reprompt
noinput	no	reprompt
nomatch	yes	reprompt
maxspechtimeout	yes	reprompt
telephone.disconnect	no	exit interpreter
all other events	yes	exit interpreter

Variables Reference

VoiceXML variables and ECMAScript variables share the same variable space. In other words, variables declared in a <script> element can be used in VoiceXML, and ECMAScript can use variables declared by a <var> element. There are 3 ways that variables can be declared:

- Declared by a <var> element
- Declared within the <script> element in an ECMAScript block
- Field item variables declared by one of form items: <block>, <field>, <initial>, <object>, <record>, <subdialog>, and <transfer>

Scope

VoiceXML defines the following variable scopes :

- **Session** - session variables are available to all VoiceXML applications within a particular session. They are declared by the interpreter and are read-only. New session variables cannot be declared by VoiceXML documents.
- **Application** - application variables are available to the root document and its application leaf documents. They are declared by the <var> elements of application root document's <vxml> element.
- **Document** - document variables are available within the VoiceXML document. They are declared by the <var> elements that are children of the <vxml> element. Certain form items (<field>, <record>, <transfer>) may declare shadow variables in addition to the field item variable itself; please refer to each tag reference for detailed information.
- **Dialog** - each dialog (<form> or <menu>) has a dialog scope that exists while the user is visiting the dialog. Variables are declared by <var> elements or by form items.
- **(Anonymous)** - <block>, <filled>, and <catch> (includes <error>, <help>, <noinput>, and <nomatch>) elements defines a new anonymous scope to contain variables declared in that element. Other predefined anonymous variables are available within the <catch> and <enumerate> elements; please refer to the tag reference for detailed information.

Session Variables

The following are standard VoiceXML session variables:

Variable	Description
session.telephone.ani	Automatic Number Identification. Provides the telephone number of the caller if the service is supported.
session.telephone.dnis	Dialed Number Identification Service. Provides the telephone number dialed by the caller if the service is supported.
session.telephone.iidigits	Information Indicator Digits. Provides information about the originating line of the caller if the service is supported.
session.telephone.uui	User to User Information. Provides information provided by ISDN call set-up if the service is supported.

The following are session variables added by VoiceGenie:

Variable	Description
session.transfer.allow	Indicates if call transfers are allowed in this session: yes or no. The interpreter throws error.telephone.noauthorization event if <transfer> is attempted and this variable is set to no.
session.com.voicegenie.callcontrol	Indicates if the VoiceGenie Call Control Extensions are enabled in this platform: true or false.

Other session variables are added to support Call Control Extensions. Please refer the to [Call Control Extension Tutorial](#) for detailed information.

Application Variables

Application variables defined in the root document can be referenced by the application leaf documents as *application.variable*.

The following are standard VoiceXML application variables:

Variable	Description
application.lastresult\$	This read-only variable holds information about the last recognition to occur within this application. It is an array of elements application.lastresult[\$i] for the N-best recognition. Please refer to the N-Best Results Tutorial (http://www.speechgenie.com/tutorials.php?tutorial=n_best) for detailed information.
application.lastresult[\$i].confidence	The confidence level for this interpretation from 0.0 (minimum confidence) to 1.0 (maximum confidence).
application.lastresult[\$i].utterance	The raw string of words that were recognized for this interpretation.
application.lastresult[\$i].inputmode	The mode in which user input was provided: dtmf or voice.
application.lastresult[\$i].interpretation	The interpretation of this result; this will be the same as rawresults for index 0 when no slot match was found, and for any index > 0.

The following are application variables added by VoiceGenie:

Variable	Description
application.lastresult[\$i].rawresults	Complete results from ASR in the format of: [?][slot1:]value1[:score1]+ [?][slot2:]value2[:score2]+... Where ? implies ambiguous match, and optional parts are quoted in [and].
application.lastresult\$.bargin	true if bargein occurred, false otherwise
application.lastresult\$.barginresult	The recognized phrase returned by ASR with barge-in. This applies only to recognition based barge-in
application.lastresult\$.audiooffset	Offset of audio data on bargein. -1 if audio was played to the end. Undefined if last played prompt was not <audio>. Please refer to the Audio Offset Features Tutorial for detailed information.
application.lastresult\$.utteranceaudio	Audio file of the user utterance.

Properties Reference

Properties are used to set values that affect platform behavior. Properties apply to their parent element and all the descendants of the parent. A property at a lower level overrides a property at a higher level.

Additional properties are supported with Call Control Extensions. Please refer to the [Call Control Tutorial](#) for information.

Speech Recognizer Properties

Property	Description	Default Value
confidencelevel	Defines threshold of speech recognition confidence level. Values range from 0.0 (minimum confidence) to 1.0 (maximum confidence). User inputs are ignored if the confidence level is below this threshold.	0.5
sensitivity	Defines sensitivity level to input. Values range from 0.0 (least sensitive to noise) to 1.0 (highly sensitive to quiet input).	0.5
speedvsaccuracy	A hint specifying the desired balance between speed vs. accuracy. Values range from 0.0 (fastest recognition) to 1.0 (best accuracy).	0.5
completetimeout	The length of silence required following user speech before the speech recognizer finalizes a result (either accepting it or throwing a nomatch event). The complete timeout is used when the speech is a complete match of an active grammar.	1s
incompletetimeout	Contrast to completetimeout, the incomplete timeout is used when the speech is an incomplete match to an active grammar.	1s
maxspeechevent	The maximum duration of user speech. If this time elapsed before the user stops speaking, the maxspeechevent event is thrown. 0 for no limit.	0
maxnbest	Maximum number of results returned by the recognizer. Also represents the maximum size of the application.lastresult[i] array.	1

DTMF Recognizer Properties

Property	Description	Default Value
interdigittimeout	The timeout period allowed between each digit when recognizing DTMF input.	3s
termtimeout	The terminating timeout to use when recognizing DTMF input.	0s
termchar	The terminating DTMF character for DTMF input recognition.	#

Prompt and Collect Properties

Property	Description	Default Value
inputmodes	Determines which input methods to use. Value is a space separated list of input methods: <ul style="list-style-type: none"> • dtmf - allows DTMF sequences as input • voice - allows voice as input 	dtmf voice
timeout	The time after which the interpreter throws a noinput event.	10s
universals	Specifies universal commands. Value is a space separated list of the following commands: (none for nothing, all for	none

	all 3 commands) <ul style="list-style-type: none"> cancel - throws the cancel event exit - throws the exit event help - throws the help event 	
asengine (VoiceGenie Extension)	Determines the ASR engine to use for this application. VoiceGenie currently supports the following engines in different platforms: <ul style="list-style-type: none"> nuance - SCO OpenServer 5.0.x and UnixWare 7.1 with Antares speechworks - RedHat Linux 6.2, Intel VPP with CSP watson - SCO OpenServer 5.0.x and UnixWare 7.1 with Antares lss - Solaris 5.x, Intel mssr - Windows 2000 and NT 4.0 (MyVoiceGenie) 	platform-specific
ttsengine (VoiceGenie Extension)	Determines the TTS engine to use for this application. VoiceGenie currently supports the following engines: <ul style="list-style-type: none"> watson ssml speechify 	platform-specific
com.voicegenie.saveutterance (VoiceGenie Extension)	Set to true to enable user utterance to be recorded. Set to false otherwise.	false
com.voicegenie.utterancedest (VoiceGenie Extension)	The directory relative to the VoiceGenie platform installation directory, in which to save user utterances.	tmp
vgasrcallog (VoiceGenie Extension)	Set to true to enable ASR engine logging. Set to false otherwise. (Available with Nuance ASR only)	false
vgasrconfidentialutterance (VoiceGenie Extension)	Set to true to disable utterance logging. Set to false to log user utterance. (Available with Nuance ASR only)	false

Prompt and Collect Properties - Barge-in

The VoiceGenie platform now supports Recognition Based Barge-in. Please refer to the tutorial for detailed information.

Property	Description	Default Value
bargein	Enables user input during prompts if set to true. No barge-in if set to false.	true
bargeintype	Specifies the barge-in type: <ul style="list-style-type: none"> energy - any noise can barge-in the prompt speech - any user utterance can barge-in the prompt recognition - only barge-in on the prompt when the user input matches a grammar 	speech
bargeinlevel (VoiceGenie Extension)	The confidence level required for barge-in to work. Values range from 0.0 (minimum confidence) to 1.0 (maximum confidence).	0.5
com.voicegenie.recognitionbargeinminimum (VoiceGenie Extension)	Minimum duration of the barge-in phrase if bargeintype=recognition	0.5s
com.voicegenie.recognitionbargeinmaximum	Maximum duration of the barge-in phrase if	1.7s

(VoiceGenie Extension) bargeintype=recognition

Fetching Properties

Property	Description	Default Value
audiofetchhint	Defines when audio files can be fetched: <ul style="list-style-type: none"> • prefetch - file can be fetched when the document is loaded • safe - only fetch when file is needed, never before 	prefetch
audiomaxage	Defines maximum acceptable age, in seconds, of cached audio resources.	undefined
audiomaxstale	Defines maximum staleness, in seconds, of expired cached audio resources.	undefined
documentfetchhint	Defines when document can be fetched: <ul style="list-style-type: none"> • prefetch - file can be fetched when the document is loaded • safe - only fetch when file is needed, never before 	safe
documentmaxage	Defines maximum acceptable age, in seconds, of cached documents.	undefined
documentmaxstale	Defines maximum staleness, in seconds, of expired cached documents.	undefined
grammarfetchhint	Defines when grammar files can be fetched: (Not supported in Speechworks ASR) <ul style="list-style-type: none"> • prefetch - file can be fetched when the document is loaded • safe - only fetch when file is needed, never before 	prefetch
grammarmaxage	Defines maximum acceptable age, in seconds, of cached grammar resources. (Not supported in Speechworks ASR)	undefined
grammarmaxstale	Defines maximum staleness, in seconds, of expired cached grammar resources. (Not supported in Speechworks ASR)	undefined
objectfetchhint	Defines when objects can be fetched: <ul style="list-style-type: none"> • prefetch - file can be fetched when the document is loaded • safe - only fetch when file is needed, never before 	prefetch
objectmaxage	Defines maximum acceptable age, in seconds, of cached object resources.	undefined
objectmaxstale	Defines maximum staleness, in seconds, of expired cached object resources.	undefined
scriptfetchhint	Defines when script files can be fetched: <ul style="list-style-type: none"> • prefetch - file can be fetched when the document is loaded • safe - only fetch when file is needed, never before 	prefetch
scriptmaxage	Defines maximum acceptable age, in seconds, of cached script resources.	undefined
scriptmaxstale	Defines maximum staleness, in seconds, of expired cached script resources.	undefined

fetchaudio	URI of audio to play while waiting for a document to be fetched.	builtin:background_audio.wav
fetchaudiodelay	The time interval to wait at the start of a fetch delay before playing the fetchaudio audio.	1s
fetchaudiominimum	Minimum time interval to play a fetchaudio source, once started, even if the fetch result arrives in the meantime.	undefined
fetchtimeout	Timeout for fetches. (Not supported in Speechworks ASR)	30s

Audio Control Properties

The Audio Control Feature is an extension to VoiceXML. Please refer to Audio Control Features Tutorial for detailed information.

Property	Description	Default Value
vgaudiocontrol	Set to true to enable Audio Control during playing of audio files. Set to false to disable the feature.	true
vgaudioskipahead	DTMF button for skipping ahead in the audio file.	undefined
vgaudioskipback	DTMF button for rewinding a few seconds in the audio file.	undefined
vgaudio softer	DTMF button for turning volume down.	undefined
vgaudiolouder	DTMF button for turning volume up.	undefined
vgaudiopause	DTMF button for pausing playback temporarily.	undefined
vgaudiostop	DTMF button for stopping the audio playback.	undefined

Miscellaneous Properties

Property	Description	Default Value
loglevel (VoiceGenie Extension)	Controls logging information that is mailed to the maintainer of the document (defined in the <meta> element). This property can only be defined in the document scope or higher (within the <vxml> element). Values are in increasing level of details: <ul style="list-style-type: none"> • 1 - Logs errors • 2 - Logs warnings • 3 - <log> elements in the document are logged • 4 - Always email log to maintainer 	1
metricslevel (VoiceGenie Extension)	Controls logging information stored in the metrics file. All <log> elements are logged to the metrics file also. This property can only be defined in the document scope or higher (within the <vxml> element). Values are in increasing level of details: <ul style="list-style-type: none"> • 0 or 1 - traces call starts and call ends • 2 - traces prompts and ASR results • 3 - traces each form item visited 	1
com.voicegenie.disablerecord (VoiceGenie Extension)	Disables the dest attribute of <record> element if the value is not undefined.	undefined
com.voicegenie.xmlencoding (VoiceGenie Extension)	An integer bit field to turn on and off encoding of attribute values and text before passing to ASR and TTS engines. This property can only be defined in the document scope or higher (within the <vxml> element). <ul style="list-style-type: none"> • bit0 - Turns on encoding for attributes in SSML • bit1 - Turns on encoding for data in SSML 	0xf

- bit2 - Turns on encoding for attributes in XML grammar
- bit3 - Turns on encoding for data in XML grammar

Platform Properties

The following VoiceGenie properties are specific to the VoiceGenie platform, and they are useful for debugging purposes.

Property	Description	Default Value
com.voicegenie.keeptmpfilestillend	Set to true to keep temporary files without being deleted during a call session. This property can only be defined in the document scope or higher (within the <vxml> element).	false
asrfiles	Saves temporary files generated for ASR processing. This property can only be defined in the document scope or higher (within the <vxml> element).	0
savetmpfiles	Saves all temporary files generated by the interpreter. This property can only be defined in the document scope or higher (within the <vxml> element).	0
disablesavetmpfiles	Disables savetmpfiles property if this property is defined. This property can only be defined in the platform defaults.vxml.	undefined
com.voicegenie.maxrecordtime	Defines the upper limit for maxtime attribute in <record> property. This property can only be defined in the platform default.vxml.	10m

<assign>

Description

The <assign> element assigns a value to a variable. A variable can be assigned without declaring it previously.

Syntax

```
<assign  
  name="string"  
  expr="ECMAScript_Expression"/>
```

Attributes

Attribute	Description
name	The name of the variable being assigned to. <i>Required</i>
expr	Value to assign to the variable. <i>Required</i>

Parents

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

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <var name="tree" expr="'an apple tree'"/>  
  <form>  
    <block>  
      <prompt> The initial value is <value expr="tree"/> </prompt>  
      <assign name="tree" expr="'apple orchards'"/>  
      <prompt> The new value is <value expr="tree"/> </prompt>  
    </block>  
  </form>  
</vxml>
```

<audio>

Description

Plays an audio file. If the file cannot be played, the content of the <audio> element is played instead. The alternate content may be text, speech markup, or another <audio> element. If both audio file and alternate content cannot be played (either no alternate content or alternate audio is not found), an error is thrown.

Syntax

```
<audio
  src="URI"
  expr="ECMAScript_Expression"
  fetchtimeout="time_interval"
  fetchhint="prefetch" | "safe"
  maxage="time_interval"
  maxstale="time_interval"
  volume="integer range from -9 to +9"
  offsetexpr="ECMAScript_Expression"
  lengthexpr="ECMAScript_Expression">
  child elements
</audio>
```

Attributes

Attribute	Description
src	The URI of the audio file. <i>Required (src or expr is required)</i>
expr	ECMAScript Expression for the URI of the audio file. If src attribute is specified, it takes precedence over this attribute. If <audio> contains this attribute that evaluates to null, this element will be skipped, including its alternate content. <i>Required (src or expr is required)</i>
fetchtimeout	The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i>
fetchhint	Defines when the audio file should be retrieved. <i>Optional</i> <ul style="list-style-type: none">• prefetch - audio file may be downloaded when the page is loaded• safe - only load the audio file when needed• stream - allows audio streaming with HTTP
maxage	Indicates the maximum time in seconds that this document will use this audio file before fetching another copy. <i>Optional</i>
maxstale	Indicates the maximum time in seconds that this document will use an audio file whose exceeded the maxage time. <i>Optional</i>
volume	Increases or decreases the volume of the audio file by the number of dB. Given as +dB or -dB: eg. "+2". <i>Optional</i>
offsetexpr	ECMAScript Expression for the time offset at which to begin playing the audio file <i>Optional (Defaults to 0)</i>
lengthexpr	ECMAScript Expression for the time length of the audio data to be played from the file. <i>Optional (Defaults to the length of referenced audio)</i>

Parents

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

Children

<audio>, <value>, <enumerate>, Speech Markup (SSML)

VoiceGenie Extensions

- Added volume attribute
- Added offsetexpr attribute - the time offset at which to begin playing the audio file
- Added lengthexpr attribute - the length of the audio data to be played from the file

Limitations/Restrictions

The child elements can only have one of <audio>, Speech Markup, or alternate text.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <block>
      <prompt>Playing the recorded message</prompt>
      <audio src="thankyou.vox">
        Reading this if thankyou.vox is not found
      </audio>
    </block>
  </form>
</vxml>
```

<block>

Description

Specifies a block of directives to be executed in document order.

Syntax

```
<block
  name="String"
  expr="ECMAScript_Expression"
  cond="ECMAScript_Expression">
  child elements
</block>
```

Attributes

Attribute	Description
name	The name of the block that can be referred within the form. <i>Optional (Defaults to an inaccessible internal variable)</i>
expr	Initial value of this block. This block will be visited only if the expression evaluates to undefined. <i>Optional (Defaults to undefined)</i>
cond	Boolean condition that must evaluate to true in order for this block to be visited. <i>Optional (Defaults to true)</i>

Parents

<form>

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <block name="first">
      <prompt>This is a block</prompt>
      <goto nextitem="second"/>
    </block>
    <block name="second">
      <prompt>This is another block</prompt>
    </block>
  </form>
</vxml>
```

<catch>

Description

Catches events thrown from the VoiceXML application or the VoiceGenie Platform. The <catch> element associates a catch with a document, dialog or form item. It contains executable content.

The catch element catches an event with the name that either matches exactly with the event attribute or there is a prefix match. A prefix match means the event attribute is a token prefix of the thrown event, where the dot is the token separator. For example, <catch event="telephone.disconnect"> is a prefix match for event telephone.disconnect.transfer.

Please refer to Event Handling for list of events and errors.

Syntax

```
<catch
  event="event1 event2 ..."
  count="Integer"
  cond="ECMAScript_Expression">
  child elements
</catch>
```

Attributes

Attribute	Description	Default
event	The event or events to catch. A space-separated list of events may be specified to catch multiple events. The empty string matches everything. <i>Required</i>	
count	The count attribute allows you to handle different occurrence of the same event differently. Each <form>, <menu> and form item maintains a counter for each event that occurs while it is being visited; these counters are reset each time the <form> or <menu> is re-entered. When there is more than one <catch> element catching the same event, it will visit the element with a smallest count that is greater or equal to the current counter. <i>Optional</i>	1
cond	Boolean condition that must evaluate to true in order for this <catch> element to catch the event. <i>Optional</i>	true

Anonymous Variables

Two anonymous variables are available within the scope of the <catch> element:

- `_event` - contains the name of the event that was thrown.
- `_message` - contains the message string from the corresponding <throw> element, or a platform defined value for events raised by the platform.

Parents

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

Children

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

VoiceGenie Extensions

VoiceGenie introduced new events for Call Control Extensions. Please refer to the [Call Control Extensions Tutorial](#) for the list of new events.

Limitations/Restrictions

- count attribute only works for the events help, noinput, or nomatch

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <grammar> goodbye </grammar>
    <catch event="goodbye">
      Thanks for using VoiceGenie, goodbye.
      <disconnect/>
    </catch>
    <field name="password">
      <prompt> what is the code word </prompt>
      <grammar> apple </grammar>
      <help> It is the name of a fruit </help>
      <catch event="noinput"> I did not hear you. </catch>
      <catch event="nomatch" count="1"> Noop. Try again </catch>
      <catch event="nomatch" count="2"> Noop. give another try </catch>
      <catch event="nomatch" count="3">
        Sorry. You didn't get it for three times. Bye
        <disconnect/>
      </catch>
    </field>
    <filled>
      <if cond="password=='goodbye' ">
        <throw event="goodbye"/>
      <else/>
        This is correct.
      </if>
    </filled>
  </form>
</vxml>
```

<choice>

Description

Defines a menu item. The <choice> element serves several purposes:

- Specifies a speech grammar fragment and/or a DTMF grammar fragment that determines when that choice has been selected
- The contents are used to form the <enumerate> prompt string
- Specifies the URI to go to when the choice is selected

Syntax

```
<choice
  dtmf="DTMF sequence"
  accept="exact " | "approximate"
  next="URI"
  event="event"
  expr="ECMAScript_Expression"
  fetchaudio="URI"
  fetchaudiodelay="time_interval"
  fetchaudiominimum="time_interval"
  fetchhint="prefetch" | "safe"
  fetchtimeout="time_interval"
  maxage="time_interval"
  maxstale="time_interval">
  child elements
</choice>
```

Attributes

Attribute	Description
dtmf	The DTMF sequence for this choice. <i>Optional</i>
accept	Overrides the setting for accept attribute in <menu>. <i>Optional (defaults to exact)</i> <ul style="list-style-type: none"> • exact - text of choice element defines the exact phrase to be recognized. • approximate - text of choice element defines an approximate recognition phrase. A subset of the words in the phrase expression can be matched. For example, "Hello world" can be matched with "Hello world", "Hello", or "World".
next	The URI of next dialog or document.
event	Throws a specified event. next and expr attributes have precedence over this attribute.
expr	An ECMAScript Expression that defines the URI to transition. next attribute has precedence over this attribute.
fetchaudio	The URI of the audio to play while waiting for the next document to be fetched.
fetchaudiodelay	The time interval to wait before playing the fetchaudio source.
fetchaudiominimum	The minimum time interval to play the fetchaudio source, even if the fetch document arrives in the meantime.
fetchtimeout	The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i>
fetchhint	Defines when the audio file should be retrieved. <i>Optional</i> <ul style="list-style-type: none"> • prefetch - audio file may be downloaded when the page is loaded • safe - only load the audio file when needed
maxage	Indicates the maximum time in seconds that this document will use this file before

maxstale	fetching another copy. <i>Optional</i> Indicates the maximum time in seconds that this document will use an file whose exceeded the maxage time. <i>Optional</i>
----------	---

Parents

<menu>

Children

<audio>, <enumerate>, <grammar>, <value>

VoiceGenie Extensions

- Attribute fetchaudiodelay has been added.
- Attribute fetchaudiominimum has been added.

Limitations/Restrictions

- Attribute fetchhint=prefetch is not supported (Optional in VoiceXML)
- <audio>, <prompt>, and Speech Markup are not supported within the <choice> element

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <menu>
    <prompt> What do you want for drink, coffee or tea? </prompt>
    <choice next="#getcoffee"> coffee </choice>
    <choice next="#gettea"> tea </choice>
    <noinput> Please say coffee or tea </noinput>
    <nomatch> Please say coffee or tea </nomatch>
  </menu>
  <form id="getcoffee">
    <block>
      <prompt>Ok, here's your coffee</prompt>
    </block>
  </form>
  <form id="gettea">
    <block>
      <prompt>Ok, here's your tea</prompt>
    </block>
  </form>
</vxml>
```

<clear>

Description

Resets one or more form items. It will do the following:

- Sets the form item variable to ECMAScript undefined.
- Reinitializes the prompt counter and event counters for the form item

Syntax

```
<clear  
  namelist="item1 item2 item3 ..." />
```

Attributes

Attribute	Description
namelist	The names of the form items to be cleared. When not specified, all of the form items in the current form are cleared. <i>Optional</i>

Parents

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

Children

None

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <form>  
    <field name="password">  
      <prompt> what is the code word </prompt>  
      <grammar> apple | reset </grammar>  
      <help> It is the name of a fruit </help>  
      <noinput> I did not hear you. </noinput>  
      <nomatch count="1"> Noop. Try again </nomatch>  
      <nomatch count="2"> Noop. give another try </nomatch>  
      <nomatch count="3">  
        Sorry. You didn't get it for three times. Bye  
        <disconnect/>  
      </nomatch>  
      <filled>  
        <if cond="password=='reset'">  
          <clear namelist="password" />  
        <else/>  
          This is correct.    </field>  
  </form>  
</vxml>
```

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

<disconnect>

Description

Disconnects the user's phone call. As a result, the interpreter context will throw the `telephone.disconnect.hangup` event. A `<disconnect>` differs from an `<exit>` in that it forces the interpreter context to drop the call.

Syntax

```
<disconnect />
```

Parents

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

Children

None

VoiceGenie Extensions

VoiceGenie Call Control Extensions adds new attributes to `<disconnect>`. Please refer to [Call Control Tutorial](#) for details.

Limitations/Restrictions

None.

<else>

Description

The <else> element is used within the conditional logic statement <if> and optional <elseif> element.

Syntax

```
<else/>
```

Parents

<if>

Children

None

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="password">
      <prompt> what is the code word </prompt>
      <grammar> apple | reset </grammar>
      <help> It is the name of a fruit </help>
      <noinput> I did not hear you. </noinput>
      <nomatch count="1"> Noop. Try again </nomatch>
      <nomatch count="2"> Noop. give another try </nomatch>
      <nomatch count="3">
        Sorry. You didn't get it for three times. Bye
        <disconnect/>
      </nomatch>
      <filled>
        <if cond="password=='reset'">
          <clear namelist="password"/>
        <else/>
          This is correct.
        </if>
      </filled>
    </field>
  </form>
</vxml>
```

<elseif>

Description

The <elseif> element is used within the conditional logic statement <if> and optional <else> element.

Syntax

```
<elseif  
  cond="ECMAScript_Expression"/>
```

Attributes

Attribute	Description
cond	Boolean expression for this conditional logic statement. <i>Required</i>

Parents

<if>

Children

None

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  ...  
  <field name="card_num" type="digits">  
    <prompt>What is your card number?</prompt>  
    <filled>  
      <if cond="card_type == 'amex' && card_num.length != 15">  
        American Express card numbers must have 15 digits.  
        <clear namelist="card_num"/>  
      <elseif cond="card_type != 'amex' && card_num.length != 16"/>  
        Mastercard and Visa card numbers have 16 digits.  
        <clear namelist="card_num"/>  
      </if>  
    </filled>  
  </field>  
</vxml>
```

<enumerate>

Description

The <enumerate> element is an automatically generated description of the choices available to the user. It specifies a template that is applied to each choice in the order they appear in the <menu> element, or in the <field> element that contains <option> elements.

Syntax

```
<enumerate>  
  child elements  
</enumerate>
```

Anonymous Variables

If <enumerate> is used with no content, it lists all choices. You can customize the content using the two anonymous variables that are available within the scope of <enumerate> element:

- `_prompt` is the prompt for the current choice
- `_dtmf` is the assigned DTMF sequence for the current choice.

Parents

<audio>, <catch>, <error>, <field>, <filled>, <help>, <if>, <menu>, <noinput>, <nomatch>, <prompt>

Children

<audio>, <value>, Speech Markup (SSML)

VoiceGenie Extensions

None.

Limitations/Restrictions

- <enumerate> element is not supported in form, document, and application level event handlers.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <menu dtmf="true">  
    <prompt>  
      Welcome home.  
      <enumerate>  
        For <value expr="_prompt"/>, press <value expr="_dtmf"/>.  
      </enumerate>  
    </prompt>  
    <choice next="http://www.example.com/vxml/sports.vxml">  
      sports  
    </choice>  
    <choice next="http://www.example.com/weather.vxml">  
      weather  
    </choice>  
    <choice next="http://www.example.com/news.vxml">  
      news
```

```
    </choice>  
  </menu>  
</vxml>
```

<error>

Description

Catches an error event. This is a shorthand notation for <catch event="error"> that catches all events of type error.

Syntax

```
<error  
  count="Integer"  
  cond="ECMAScript_Expression">  
  child elements  
</error>
```

Attributes

Attribute	Description
count	The event count as in <catch> element. <i>Optional</i>
cond	Boolean condition to test to see if the event is caught by this <error>. <i>Optional</i>

Anonymous Variables

Two anonymous variables are available within the scope of the <catch> element:

- `_event` - contains the name of the event that was thrown.
- `_message` - contains the message string from the corresponding <throw> element, or a platform defined value for events raised by the platform.

Parents

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

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

- count attribute does not work for error events

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <error>  
    <prompt>An error has occurred.</prompt>  
    <exit/>  
  </error>  
  <form>  
    <block>  
      <audio src="badaudio.vox"/>  
    </block>
```

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

<exit>

Description

Exit the interpreter session. Returns control to the interpreter context, which determines what to do next. This element differs from <return> element in that <exit> terminates all loaded elements while <return> returns from a <subdialog> invocation.

Syntax

```
<exit  
  expr="ECMAScript_Expression"  
  namelist="item1 item2 item3..." />
```

Attributes

Attribute	Description
expr	The expression to be evaluated and returned. <i>Optional</i>
namelist	Variable names to be returned to interpreter context. Returns nothing by default. <i>Optional</i>

Parents

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

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

- Attributes expr and namelist are not supported.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <error>  
    <prompt>An error has occurred.</prompt>  
    <exit/>  
  </error>  
  <form>  
    <block>  
      <audio src="badaudio.vox" />  
    </block>  
  </form>  
</vxml>
```

<field>

Description

Declares an input field in a form. Prompts user for values that match a grammar.

Syntax

```
<field
  name="String"
  expr="ECMAScript_Expression"
  cond="ECMAScript_Expression"
  type="boolean" | "currency" | "phone" | "number" | "digit" | "time" | "date"
  slot="String"
  modal="true" | "false"
  saveutterance="true" | "false">
  child elements
</field>
```

Attributes

Attribute	Description
name	The name of field item variable that holds the matched user input. <i>Required</i>
expr	Initial value of this field item variable. This field will be visited only if the expression evaluates to undefined. <i>Optional (Defaults to undefined)</i>
cond	Boolean condition that must evaluate to true in order for this field to be visited. <i>Optional (Defaults to true)</i>
type	Specifies a built-in grammar. <i>Optional (as an alternative to <grammar> element)</i>
slot	The name of the grammar slot used to populate the field item variable for mixed initiative dialog. <i>Optional (Defaults to the variable name)</i>
modal	Set to true if only the field's grammars are enabled. Otherwise all active grammars are enabled. <i>Optional (Defaults to false)</i>
saveutterance	Set to true to enable recording of user utterance. Please refer to the Saved Utterance Tutorial for detailed information. <i>Optional (Defaults to false)</i>
utterancedest	Specifies the relative directory name to save utterance audio files. <i>Optional</i>
endbeep	Set to true to play a beep after the prompt. Set to false otherwise. <i>Optional (Defaults to false)</i>

Shadow Variables

The <field> shadow variable (name\$) has the following properties after the field is filled by user input: (These properties are available in object application.lastresult\$ as well)

- name\$.confidence - recognition confidence level. Floating point value between 0 to 1.0.
- name\$.utterance - raw string of words that were recognized.
- name\$.bargainscore - confidence level of the bargain. Floating point value between 0 to 1.0.
- name\$.rawresults - complete result from ASR in the format of:
[?][slot1:]value1[:score1]+ [?][slot2:]value2[:score2]+...
Where ? implies ambiguous match, and optional parts are quoted in [and].
- name\$.interpretation - interpretation for this result.

The following shadow variables are available within the field's <nomatch> also:

- name\$.inputmode - user input type. Either dtmf or voice.
- name\$.bargain - true if bargain occurred, false otherwise.

- name\$.bargeinphrase - the recognized phrase returned by ASR with barge-in. This applies only to recognition based barge-in.
- name\$.audiooffset - offset of audio data on bargein. -1 if audio was played to the end. Undefined if last played prompt was not <audio>. Please refer to the Audio Offset Features Tutorial for detailed information.
- name\$.utteranceaudio - audio file of the user utterance, relative to the VoiceGenie software installation directory on the VoiceGenie platform.

Built-in Types

The following built-in types are supported:

Type	Description
boolean	Grammar for affirmative and negative phrases. Returns "true" for yes and "false" for no.
date	Grammar for specifying a date. Returns a fixed-length date string format yyyyymmdd. If the year is not specified, yyyy becomes ????, mm for month becomes ??, and dd for day becomes ??.
digits	Grammar for a string of digits from spoken or DTMF input. Returns the string of digits.
currency	Grammar for currency amount. The result is a string with the format UUUm.n, where UUUm is the three character currency indicator according to ISO standard 4217:1995, or m.n if the currency is not spoken by the user or if the currency cannot be reliably determined (e.g. "dollar" and "peso" are ambiguous). If the field value is subsequently used in a prompt, it will be spoken as a currency amount appropriate to the current locale. The set of accepted spoken currency formats is platform dependent and may vary by locale.
number	Grammar for numbers. Returns a string of digits from 0 to 9, and may optionally include a decimal point (".") and/or a plus or minus sign.
phone	Grammar for a phone number. DTMF asterisk "*" represents "x". The result is a string containing a telephone number consisting of a string of digits and optionally containing the character "x" to indicate a phone number with an extension. For North America, a result could be "8005551234x789".
time	Grammar for specifying time. The result is a five character string in the format hhmmx, where x is one of "a" for AM, "p" for PM, "h" to indicate a time specified using 24 hour clock, or "?" to indicate an ambiguous time. Input can be via DTMF. Because there is no DTMF convention for specifying AM/PM, in the case of DTMF input, the result will always end with "h" or "?".

The following built-in types can be parameterized with this syntax:

```
typename?parameter1=value1;parameter2=value2
```

For example, we can assign DTMF sequence for a Boolean type:

```
<field name="mychoice" type="boolean?y=5;n=6">
```

Type	Input Parameters
boolean	<ul style="list-style-type: none"> • y - the DTMF sequence for an affirmative answer • n - the DTMF sequence for a negative answer
digits	<ul style="list-style-type: none"> • minlength - minimum number of digits • maxlength - maximum number of digits • length - exact number of digits <p>If there is a conflict among these parameters, a <code>error.badfetch</code> event is thrown.</p>

Parents

<form>

Children

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

VoiceGenie Extensions

- Added Saved Utterance feature. Includes added attributes (saveutterance, utterancedest) and utteranceaudio shadow variable.
- Added endbeep attribute.
- Added shadow variables bargein, bargeinphrase, audiooffset, which are also available in application.lastresult even if a nomatch event occurred.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form id="testfield">
    <block>
      Welcome to VoiceGenie restaurant.
    </block>

    <!-- field using built in grammar with digits type -->
    <field name="built_in_gram1" type="digits">
      <prompt> How many people you have? </prompt>
      <filled>
        Here is a table for <value expr="built_in_gram1"/> people.
        Have a seat.
        <goto nextitem="built_in_gram2" />
      </filled>
    </field>

    <!-- filed using built in grammar with boolean type -->
    <field name="built_in_gram2" type="boolean">
      <prompt> Are you ready for ordering now ? </prompt>
      <filled namelist="built_in_gram2">
        <if cond="built_in_gram2 == 'true'">
          <goto nextitem="optionlist" />
        <elseif cond="built_in_gram2 == 'false'"/>
          <goto nextitem="notready" />
        </if>
      </filled>
    </field>

    <!-- filed using option list -->
    <field name="optionlist">
      <prompt>
        Which entree would you like, baked potato, french fries?
      </prompt>
      <option value="baked potato"> baked potato </option>
      <option value="french fries"> french fries</option>
      <filled>
        ok. <value expr="optionlist"/> as entree.
```

```
<goto nextitem="explicit_graml" />
</filled>
</field>

<!-- filled using explicit grammar -->
<field name="explicit_graml">
  <prompt>
    Here is the main dishes you could choose from,
    Vegetarain delight, Prime Rib, Baked clams
  </prompt>
  <grammar type="application/x-abnf">
    Vegetarain delight |
    Prime Rib |
    Baked clams
  </grammar>
  <filled>
    Good choice.
    <value expr="explicit_graml"/>
    is my favorite as well.
    <goto nextitem="ordertaken" />
  </filled>
</field>

<block name="goodbye">
  good bye
  <disconnect/>
</block>
<block name="ordertaken">
  Please wait. Your meal will be ready shortly.
  <disconnect/>
</block>
<block name="notready">
  Ok. Take your time and call back later.
  <disconnect/>
</block>
</form>
</vxml>
```

<filled>

Description

Specifies an action to perform when some combination of fields are filled by user input. It may occur as a child of the <form> element, or as a child of a field item.

Syntax

```
<filled
  mode="all" | "any"
  namelist="item1 item2 item3...">
  child elements
</filled>
```

Attributes

Attribute	Description
mode	This attribute cannot be defined when <filled> is in a field item. There are 2 filled modes: <i>Optional (Default is all)</i> <ul style="list-style-type: none">all - this action is executed when all of the fields in namelist attribute are filled, and at least one has been filled by the last user input.any - this action is executed when any of the specified fields is filled by the last user input
namelist	This attribute cannot be defined when <filled> is in a field item. The namelist defines the fields to trigger on. <i>Optional (Default is list of all form's field items)</i>

Parents

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

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  ...
  <field name="card_num" type="digits">
    <prompt>What is your card number?</prompt>
    <filled>
      <if cond="card_type == 'amex' && card_num.length != 15">
        American Express card numbers must have 15 digits.
        <clear namelist="card_num"/>
      <elseif cond="card_type != 'amex' && card_num.length != 16"/>
```

```
Mastercard and Visa card numbers have 16 digits.  
<clear namelist="card_num" />  
</if>  
</filled>  
</field>  
</vxml>
```

<form>

Description

Forms are key components in VoiceXML applications. Forms collect user input and present information to the user. A form can define a field item (for collecting user input) or a control item (contains procedural items to help collect inputs).

Syntax

```
<form
  id="String"
  scope="dialog" | "document"
  cleardtmf="true" | "false">
  child elements
</form>
```

Attributes

Attribute	Description
id	Defines the name of the form. If specified, it can be referenced within the same document or from another document. eg. <form id="hello"> can be referenced as <goto next="#hello"/>. <i>Optional</i>
scope	Defines the scope of the form's grammar. <i>Optional (Defaults to dialog)</i> <ul style="list-style-type: none">• dialog - form's grammars are only active within the form• document - form's grammars are active throughout the document. If this is an application root document, then the form grammars are active throughout the application.
cleardtmf	On a form entry, the DTMF buffer is not cleared (ie. carries DTMF inputs from previous forms). Set to true to clear the buffer. <i>Optional (Defaults to false)</i>

Parents

<vxml>

Children

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

VoiceGenie Extensions

- Added cleardtmf property
- <script> can be a child of <form>

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form id="first">
    <block>
      <prompt>Hi, this is the first form</prompt>
```

```
    <goto next="#second"/>
  </block>
</form>
<form id="second">
  <block>
    <prompt>Thank you for coming to the second form</prompt>
  </block>
</form>
</vxml>
```

<goto>

Description

Transitions to another item in the current form, another dialog in the same document, or to a different document. Transitioning to another dialog in the current document or to a different document will cause the old dialog's variables to be lost. Document variables are retained when transitioning to the same document with empty URI reference. For example, if the following <goto> elements behaves differently if the current document has an URI of `http://www.voicegenie.com/index.vxml`:

Document variables are retained.

```
<goto next="#foo"/>
```

Document variables are not retained.

```
<goto next=http://www.voicegenie.com/index.vxml#foo/>
```

Syntax

```
<goto
  next="URI"
  expr="ECMAScript_Expression"
  nextitem="String"
  expritem="ECMAScript_Expression"
  fetchaudio="URI"
  fetchaudiodelay="time_interval"
  fetchaudiominimum="time_interval"
  fetchhint="prefetch" | "safe"
  fetchtimeout="time_interval"
  maxage="time_interval"
  maxstale="time_interval"/>
```

Attributes

Attribute	Description
next	The URI in which to transition to. <i>Required (One of next, expr, nextitem, expritem is required)</i>
expr	ECMAScript Expression that yields the URI. <i>Required (One of next, expr, nextitem, expritem is required)</i>
nextitem	The name of the next form item to visit in the current form. <i>Required (One of next, expr, nextitem, expritem is required)</i>
expritem	ECMAScript Expression that yields the name of the next form item to visit. <i>Required (One of next, expr, nextitem, expritem is required)</i>
fetchaudio	The URI of audio to play while waiting for the next document to be fetched.
fetchaudiodelay	The time interval to wait before playing the fetchaudio source.
fetchaudiominimum	The minimum time interval to play the fetchaudio source, even if the fetch document arrives in the meantime.
fetchtimeout	The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i>
fetchhint	Defines when the audio file should be retrieved. <i>Optional</i> <ul style="list-style-type: none"> • prefetch - audio file may be downloaded when the page is loaded • safe - only load the audio file when needed
maxage	Indicates the maximum time in seconds that this document will use this file before fetching another copy. <i>Optional</i>
maxstale	Indicates the maximum time in seconds that this document will use the file whose

exceeded the maxage time. *Optional*

Parents

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

Children

None.

VoiceGenie Extensions

- Attribute fetchaudiodelay is added.
- Attribute fetchaudiominimum is added.

Limitations/Restrictions

- Attribute fetchhint=prefetch is not supported.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form id="maindish">
    <field name="done" type="boolean">
      <prompt> Are you done here ? </prompt>
      <filled>
        <if cond="done== 'FALSE'">
          <goto nextitem="notdone" />
        </if>
      </filled>
    </field>

    <field name="whatnext">
      <prompt>
        Would like any desserts or you want me to bring your bill?
      </prompt>
      <grammar type="application/x-abnf">
        bill|
        desserts
      </grammar>
      <filled>
        <if cond="whatnext=='bill'">
          <goto nextitem="bill" />
        <elseif cond="whatnext=='desserts'" />
          <goto next="#desserts" />
        </if>
      </filled>
    </field>
    <block name="bill">
      Please wait. I am getting your bill for you.
      <disconnect/>
    </block>
    <block name="notdone">
      Oh, I am sorry. Enjoy.
      <disconnect/>
    </block>
  </form>
</vxml>
```

```
</form>
<form id="desserts">
  <field name="choice">
    <prompt>
      What would you like for your desserts?
    </prompt>
    ...
  </field>
</form>
</vxml>
```

<grammar>

Description

Specifies a grammar for speech recognition. When the grammar is recognized, the application may perform an action (such as transition to another field item), or assign values to a field item variable (multiple values can also be assigned with mixed initiative forms).

Please refer to the following tutorials on how to construct different types of grammar:

http://www.speechgenie.com/tutorials.php?tutorial=grammar_howto

http://www.speechgenie.com/tutorials.php?tutorial=osr_sg

Note

Inline grammars that use the XML grammar format will be passed into the ASR Engine as an XML document. The interpreter first decodes the VoiceXML document, and re-encodes the inline grammar into an XML document to pass to the ASR Engine. Please refer to the [Character Encoding/Decoding Tutorial](#) for detailed information with examples.

Syntax

```
<grammar
  xml:lang="language"
  src="URI"
  expr="ECMAScript_Expression"
  scope="dialog" | "document"
  type="MIME_Type"
  mode="dtmf" | "voice"
  root="String"
  version="version_number"
  weight="positive floating_point"
  fetchhint="prefetch" | "safe"
  fetchtimeout="time_interval"
  maxage="time_interval"
  maxstale="time_interval">
  inline grammar
</grammar>
```

Attributes

Attribute	Description
xml:lang	The language and locale identifier of the grammar. Please refer to the Multi-Language support Tutorial for detailed information. VoiceGenie supports the following values: <i>Optional (Defaults to en-US)</i> <ul style="list-style-type: none">• en-US• en-UK• fr-fr• de-de
src	URI of the grammar specification. The URI can be of the following formats: <ul style="list-style-type: none">• External grammar file: URL of the grammar file• Built-in grammars: builtin:grammar/type (refer to <field> element for built-in types)
expr	ECMAScript Expression that yields the URI. <i>Optional</i>
scope	Defines the scope of the grammar. This attribute can be defined <i>only</i> if this

	<p><grammar> element is the child of <form> or <menu> element. <i>Optional (Defaults to dialog)</i></p> <ul style="list-style-type: none"> • dialog - grammar is only active within the form • document - grammar is active throughout the document. If this is an application root document, then the grammar is active throughout the application.
type	<p>Defines the MIME type of grammar format. The following types are supported:</p> <ul style="list-style-type: none"> • application/x-abnf for Nuance ASR • application/x-voicegenie-nuance for Nuance ASR • application/x-voicegenie-watson for Watson ASR • application/grammar-xml for Speechworks ASR • application/x-swi-grammar-compiled for Speechworks ASR • application/x-jsgf for LSS ASR • application/x-ms-xml for MSSR ASR <p>Default values depend on the ASR Engine and whether it is an inline grammar or external grammar:</p> <ul style="list-style-type: none"> • Nuance - application/x-abnf for inline and application/x-voicegenie-nuance for external • Speechworks - application/grammar-xml for inline and external • Watson - simple ABNF for inline and application/x-voicegenie-watson for external • LSS - application/x-jsgf for both inline and external • MSSR - simple ABNF for inline and application/x-ms-xml for external
mode	<p>Defines the mode of the grammar: <i>Required</i></p> <ul style="list-style-type: none"> • voice - voice input • dtmf - DTMF input (This replaces the obsolete <dtmf> element in VoiceXML 1.0)
root	<p>Specifies the root rule of the grammar when this is an inline XML grammar. <i>Optional (Defaults to the default rule)</i></p>
version	<p>Defines the version of the grammar. <i>Optional (Defaults to 1.0)</i></p>
weight	<p>Defines the weight of a grammar, which indicates the occurrence likelihood of the grammar. Grammar weights only affect grammar processing; they do not affect the processing of grammar results. Different ASR engines and VoiceXML platforms also treat weights differently. This attribute does not apply for DTMF grammars and implicit grammars (ie. <grammar> elements as children of <option> or <choice> tags). The range of values depends on the ASR Engine, but it is usually 0.0 to 1.0. <i>Optional (Defaults to 1.0)</i></p>
fetchtimeout	<p>The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i></p>
fetchhint	<p>Defines when the audio file should be retrieved. <i>Optional</i></p> <ul style="list-style-type: none"> • prefetch - audio file may be downloaded when the page is loaded • safe - only load the audio file when needed
maxage	<p>Indicates the maximum time in seconds that this document will use the grammar file before fetching another copy. <i>Optional</i></p>
maxstale	<p>Indicates the maximum time in seconds that this document will use a grammar file whose exceeded the maxage time. <i>Optional</i></p>

Parents

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

Children

None.

VoiceGenie Extensions

- Added expr attribute

Limitations/Restrictions

- Speechworks ASR does not support maxage and maxstale attributes.

<help>

Description

Catches a help event. This is a shorthand notation for <catch event="help">.

Syntax

```
<error  
  count="Integer"  
  cond="ECMAScript_Expression">  
  child elements  
</error>
```

Attributes

Attribute	Description
count	The event count as in <catch> element. <i>Optional</i>
cond	Boolean condition to test to see if the event is caught by this <help>. <i>Optional</i>

Parents

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

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <form>  
    <field name="hello">  
      <grammar>hello</grammar>  
      <help>Just say hello</help>  
      <prompt>Say hello</prompt>  
      <noinput>Say something</noinput>  
      <filled>  
        <prompt>Hello, world!</prompt>  
      </filled>  
    </field>  
  </form>  
</vxml>
```

<if>

Description

Defines if-then-else conditional logic.

Syntax

```
<if  
  cond="ECMAScript_Expression">  
  child elements  
</if>
```

Attributes

Attribute	Description
cond	Boolean expression for this conditional logic statement. <i>Required</i>

Parents

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

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  ...  
  <field name="card_num" type="digits">  
    <prompt>What is your card number?</prompt>  
    <filled>  
      <if cond="card_type == 'amex' && card_num.length != 15">  
        American Express card numbers must have 15 digits.  
        <clear namelist="card_num"/>  
      <elseif cond="card_type != 'amex' && card_num.length != 16"/>  
        Mastercard and Visa card numbers have 16 digits.  
        <clear namelist="card_num"/>  
      </if>  
    </filled>  
  </field>  
</vxml>
```

<initial>

Description

Declares initial logic upon entry into a (mixed-initiative) form.

Unlike <field>, <initial> has no grammars and no <filled> action. <initial> can request user input and handle events. <initial> continues to be visited while its form item variable is undefined and condition is true. When any of the form's fields are filled by user input, then all <initial> form item variables are set to true, before any <filled> actions are executed. Then the form will visit those fields that are still unfilled to complete the form.

Syntax

```
<initial
  name="String"
  expr="ECMAScript_Expression"
  cond="ECMAScript_Expression">
  child elements
</initial>
```

Attributes

Attribute	Description
name	The name of form item variable used. Use this variable if you want to explicitly control <initial> execution. <i>Optional (Defaults to an inaccessible internal variable)</i>
expr	Initial value of this form item variable. <initial> will be visited only if the expression evaluates to undefined. <i>Optional (Defaults to undefined)</i>
cond	Boolean condition that must evaluate to true in order for this <initial> element to be visited. <i>Optional (Defaults to true)</i>

Parents

<form>

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
<form id="get_from_and_to_cities">
  <grammar src="from_to.grammar"/>
  <block>
    Welcome to the Driving Directions By Phone.
  </block>
  <initial name="bypass_init">
```

```
<prompt>Where do you want to drive from and to?</prompt>
<nomatch count="1">
  Please say something like this,
  "from Toronto Ontario to Ottawa Ontario".
</nomatch>
<nomatch count="2">
  I'm sorry, I still don't understand.
  I'll ask you for information one piece at a time.
  <assign name="bypass_init" expr="true"/>
  <reprompt/>
</nomatch>
</initial>
<field name="from_city">
  <grammar src="city.grammar"/>
  <prompt>From which city are you leaving?</prompt>
  ...
</field>
<field name="to_city">
  <grammar src="city.grammar"/>
  <prompt>Which city are you going to?</prompt>
  ...
</field>
  ...
</form>
</vxml>
```

<link>

Description

A <link> element may have one or more grammars. The grammars are in scope of the element containing this <link> element. When the user input matches one of the linked grammars, it activates the <link> to either throw an event, or transition to another document or dialog.

Syntax

```
<link
  next="URI "
  expr="ECMAScript_Expression"
  event="event "
  dtmf="DTMF Sequence"
  fetchaudio="URI "
  fetchhint="prefetch" | "safe"
  fetchtimeout="time_interval"
  maxage="time_interval"
  maxstale="time_interval">
  child_elements
</link>
```

Attributes

Attribute	Description
next	The URI to transition to when a match is recognized. <i>Required (One and only one of next, expr, or event is required)</i>
expr	Expression that yields the URI to transition to when a match is recognized. <i>Required (One and only one of next, expr, or event is required)</i>
event	The event to throw when the user input matches one of the linked grammars. <i>Required (One and only one of next, expr, or event is required)</i>
dtmf	The DTMF sequence for this link. It is equivalent to a simple DTMF <grammar>. This attribute can be used at the same time with other grammars: the link is activated when user input matches a linked grammar or this DTMF sequence. <i>Optional</i>
fetchaudio	The URI of audio to play while waiting for the next document to be fetched. <i>Optional</i>
fetchtimeout	The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i>
fetchhint	Defines when the audio file should be retrieved. <i>Optional</i> <ul style="list-style-type: none">• prefetch - audio file may be downloaded when the page is loaded• safe - only load the audio file when needed
maxage	Indicates the maximum time in seconds that this document will use this file before fetching another copy. <i>Optional</i>
maxstale	Indicates the maximum time in seconds that this document will use the file whose exceeded the maxage time. <i>Optional</i>

Parents

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

Children

<grammar>

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <link event="goodbye" dtmf="123"/>
  <catch event="goodbye">
    <prompt>Thank you for trying VoiceGenie, goodbye.</prompt>
    <exit/>
  </catch>
  <form>
    <field name="hello">
      <grammar>hello</grammar>
      <prompt>Say hello</prompt>
      <noinput>Say something</noinput>
      <filled>
        <prompt>Hello, world!</prompt>
      </filled>
    </field>
  </form>
</vxml>
```

<log>

Description

The <log> element allows the application to generate debug messages. This element can contain combination of text and <value> elements, which the resulting message will be the concatenation of the text and string value of <value> element. The messages will be logged to the metrics file for viewing.

Syntax

```
<log
  expr="ECMAScript_Expression"
  label="String"
  cond="ECMAScript_Expression"
  level="Integer">
  text
</log>
```

Attributes

Attribute	Description
expr	A string expression that will be appended to the <log> element content and label. <i>Optional</i>
label	A string label that will be appended to the <log> element content before expr. <i>Optional</i>
cond	Boolean expression that must evaluate to true for this log to be executed. <i>Optional (Defaults to true)</i>
level	An integer value that must be less than or equal to property value loglevel for this log to be executed. <i>Optional (Defaults to 1)</i>

Parents

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

Children

<value>

VoiceGenie Extensions

- Added attributes cond and level.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="hello">
      <grammar>hello | world</grammar>
      <prompt>Say hello</prompt>
      <noinput>Say something</noinput>
      <filled>
```

```
<log>The user said <value expr="hello" />.</log>  
<prompt>Hello, world!</prompt>  
</filled>  
</field>  
</form>  
</vxml>
```

<menu>

Description

Presents a menu of choices to the user and transitions to another dialog based on user input.

Syntax

```
<menu
  id="String"
  scope="dialog" | "document"
  dtmf="true" | "false"
  accept="exact" | "approximate">
  child elements
</menu>
```

Attributes

Attribute	Description
id	Defines the name of the form. If specified, it can be referenced within the same document or from another document. eg. <form id="hello"> can be referenced as <goto next="#hello"/>. <i>Optional</i>
scope	Defines the scope of the form's grammar. <i>Optional (Defaults to dialog)</i> <ul style="list-style-type: none">• dialog - form's grammars are only active within the form• document - form's grammars are active throughout the document. If this is an application root document, then the form grammars are active throughout the application.
dtmf	Enables DTMF input for all menu choices. <i>Optional (Defaults to false)</i> <ul style="list-style-type: none">• true - assigns DTMF sequence "1", "2", "3"... to the choices in document order for all menu choices that do not explicitly define a DTMF sequence.• false - interpreter does not implicitly assign DTMF sequences
accept	Specifies the default grammar for each <choice> element. <i>Optional (defaults to exact)</i> <ul style="list-style-type: none">• exact - text of choice element defines the exact phrase to be recognized.• approximate - text of choice element defines an approximate recognition phrase. A subset of the words in the phrase expression can be matched. For example, "Hello world" can be matched with "Hello world", "Hello", or "World".

Parents

<vxml>

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
<menu dtmf="true">
  <prompt>
    What information do you want?
    <enumerate>
      For <value expr="_prompt"/>, press <value expr="_dtmf"/>
    </enumerate>
  </prompt>
  <choice next="#exit"> goodbye <choice>
  <choice next="sports.vxml"> sports </choice>
  <choice next="weather.vxml"> weather </choice>
  <choice next="stock.vxml"> stock </choice>
</menu>
<form id="exit">
  <block>
    Good bye.
    <disconnect/>
  </block>
</form>
</vxml>
```

<meta>

Description

Specifies general information about the VoiceXML application.

Syntax

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

Attributes

Attribute	Description
name	Name of the meta-data property. <i>Required (either name or http-equiv is required)</i>
content	Value of the meta-data property. <i>Required</i>
http-equiv	Name of an HTTP response header. <i>Required (either name or http-equiv is required)</i>

Meta-data Properties

Name	Description
application	Defines name of the application for billing purposes.
callrequest	If value is <code>decline</code> , then the application will decline incoming calls. Otherwise, the application accepts incoming calls.
maintainer	Email address that log files will be sent to.

Parents

<vxml>

Children

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="hello">
      <grammar>hello</grammar>
      <prompt>Say hello</prompt>
      <nomatch>That's not hello</nomatch>
      <filled>
        <prompt>Hello, world!</prompt>
      </filled>
    </field>
  </form>
</vxml>
```

<noinput>

Description

Handles the event when the user does not give inputs in a field. This is a shorthand notation for <catch event="noinput"> that catches all events of type error.

Syntax

```
<noinput
  count="Integer"
  cond="ECMAScript_Expression">
  child elements
</noinput>
```

Attributes

Attribute	Description
count	The event count as in <catch> element. <i>Optional</i>
cond	Boolean condition to test to see if the event is caught by this <noinput>. <i>Optional</i>

Parents

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

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="hello">
      <grammar>hello</grammar>
      <prompt>Say hello</prompt>
      <noinput>Say something</noinput>
      <filled>
        <prompt>Hello, world!</prompt>
      </filled>
    </field>
  </form>
</vxml>
```

<nomatch>

Description

Handles the event when the user gives unrecognized input. This is a shorthand notation for <catch event="nomatch"> that catches all events of type error.

Syntax

```
<nomatch
  count="Integer"
  cond="ECMAScript_Expression">
  child elements
</nomatch>
```

Attributes

Attribute	Description
count	The event count as in <catch> element. <i>Optional</i>
cond	Boolean condition to test to see if the event is caught by this <nomatch>. <i>Optional</i>

Parents

<field>, <form>, <initial>, <menu>, <object>, <record>, <subdialog>, <transfer>, <v xml>

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="hello">
      <grammar>hello</grammar>
      <prompt>Say hello</prompt>
      <nomatch>That's not hello</nomatch>
      <filled>
        <prompt>Hello, world!</prompt>
      </filled>
    </field>
  </form>
</vxml>
```

<object>

Description

Interact with custom extension. A VoiceXML platform may have platform-specific functionality that an application wants to use, such as speaker verification, native components, additional telephony, etc. Such platform-specific objects are accessed using the <object> element, which is analogous to the HTML <OBJECT> element.

Syntax

```
<object
  name="String"
  expr="ECMAScript_Expression"
  cond="ECMAScript_Expression"
  classid="URI"
  codebase="URI"
  codetype=""
  data="URI"
  type=""
  archive="URI"
  fetchhint="prefetch" | "safe"
  fetchtimeout="time_interval"
  maxage="time_interval"
  maxstale="time_interval">
  child elements
</object>
```

Attributes

Attribute	Description
name	Defines the name of the variable when the object is evaluated. <i>Required</i>
expr	Initial value of the variable. <i>Optional (Defaults to undefined)</i>
cond	Boolean expression that must evaluate to true in order for this object to execute. <i>Optional (Defaults to true)</i>
classid	The URI specifying the location of the object's implementation.
codebase	The base path used to resolve relative URIs specified by classid, data, and archive. <i>Optional (Defaults to base URI of current document)</i>
codetype	The content type of data expected when downloading the object specified by classid. <i>Optional (Defaults to the value of the type attribute)</i>
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.
type	The content type of the data specified by the data attribute.
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.
fetchtimeout	The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i>
fetchhint	Defines when the audio file should be retrieved. <i>Optional</i> <ul style="list-style-type: none"> • prefetch - audio file may be downloaded when the page is loaded • safe - only load the audio file when needed
maxage	Indicates the maximum time in seconds that this document will use this object before fetching another copy. <i>Optional</i>
maxstale	Indicates the maximum time in seconds that this document will use an object whose

exceeded the maxage time. *Optional*

Parents

<vxml>

Children

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

VoiceGenie Extensions

None.

Limitations/Restrictions

No VoiceGenie platform-specific objects are currently defined.

<option>

Description

Specifies an option in a <field>. It is a convenient way to specify a list of choices in a field without specifying a grammar. The grammar is generated automatically from the text contained in each <option>.

Syntax

```
<option
  dtmf="DTMF sequence"
  value="String">
  text
</option>
```

Attributes

Attribute	Description
dtmf	DTMF sequence for this option. <i>Optional</i>
value	The string to assign to the field item variable this option is selected, whether by speech or DTMF. <i>Optional (Defaults to dtmf attribute if specified. Otherwise, defaults the option text with leading and trailing white space removed.)</i>

Parents

<field>

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="maincourse">
      <prompt>
        Please select an entree. Today, we're featuring:
        <enumerate/>
      </prompt>
      <option dtmf="1" value="fish"> swordfish </option>
      <option dtmf="2" value="beef"> roast beef </option>
      <option dtmf="3" value="chicken"> frog legs </option>
      <filled>
        <prompt>
          <value expr="maincourse"/>, good choice.
          Please enjoy your meal.
        </prompt>
```

```
<submit next="maincourse.cgi"  
        method="post" namelist="maincourse"/>  
</filled>  
</field>  
</form>  
</vxml>
```

<param>

Description

Specifies values that are passed into <object> or in <subdialog>. When <param> is contained in a <subdialog> element, the values specified are used to initialize <var> declarations in the subdialog that is invoked. The initialization takes precedence over the expr attribute in <var>.

Syntax

```
<param
  name="String"
  value="String"
  expr="ECMAScript_Expression"
  valuetype="data" | "ref"
  type="MIME_type" />
```

Attributes

Attribute	Description
name	The name associated with the parameter. <i>Required</i>
value	String value of the parameter. <i>Required (One of value or expr is required)</i>
expr	Expression that yields the parameter value. <i>Required (One of value or expr is required)</i>
valuetype	Indicates the reference type of the value. This attribute is not used for <subdialog>. <i>Optional (Defaults to data)</i> <ul style="list-style-type: none">• data - value is actual data• ref - value referenced by an URI.
type	The media type of the result provided by a URI if the valuetype is ref; only relevant for uses of <param> in <object>. <i>Optional</i>

Parents

<object>, <subdialog>

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

- Value in expr attribute cannot be an object.

Example

Document that calls the subdialog:

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <subdialog name="result" src="subdialog.vxml">
      <param name="birthday" expr="'2000-02-10'"/>
      <param name="age" value="100"/>
    </subdialog>
  </form>
</vxml>
```

```
<filled>  
  <submit next="http://myservice.example.com/cgi-bin/process"/>  
</filled>  
</subdialog>  
</form>  
</vxml>
```

Document containing the subdialog:

```
<?xml version="1.0"?>  
<vxml version="2.0">  
<form id="getdriverslicense">  
  <var name="birthday"/>  
  <var name="age"/>  
  <block>  
    <prompt>  
      Hello, your birthday is <value expr="birthday"/>  
      and you are <value expr="age"/> years old.  
    </prompt>  
    <return/>  
  </block>  
</form>  
</vxml>
```

<prompt>

Description

Plays TTS to the user.

Note

Any prompts that contain Speech Markup (SSML) will be passed into the TTS Engine as an XML document. The interpreter first decodes the VoiceXML document, and re-encodes the prompt content into an XML document to pass to the TTS Engine. Please refer to the [Character Encoding/Decoding Tutorial](#) for detailed information with examples.

Syntax

```
<prompt
  bargein="true" | "false"
  bargeintype="energy" | "speech" | "recognition"
  cond="ECMAScript_Expression"
  count="Integer"
  timeout="time_interval">
  child elements
</prompt>
```

Attributes

Attribute	Description
bargein	Enables user input during the prompt: <i>Optional</i> <ul style="list-style-type: none"> true - user input can barge-in the current prompt. Subsequent prompts in the queue will be treated as bargein=true false - user barge-in to the current prompt until the prompt has finished playing
bargeintype	Specifies the barge-in type: <ul style="list-style-type: none"> energy - any noise can barge-in the prompt speech - any user utterance can barge-in the prompt recognition - only bargin-in the prompt when the user input matches a grammar
cond	Boolean expression that must evaluate to true for this prompt to be played. <i>Optional (Defaults to true)</i>
count	Each field item maintains a counter of number of times the item has been visited. The count attribute allows the application to play different prompts based on the counter. The prompt will be played when the counter reaches the count attribute. <i>Optional (Defaults to 1)</i>
timeout	Time to wait before throwing a noinput event. <i>Optional</i>

Parents

<block>, <catch>, <choice>, <error>, <field>, <filled>, <if>, <initial>, <menu>, <noinput>, <nomatch>, <object>, <prompt>, <record>, <subdialog>

Children

<audio>, <enumerate>, <value>, Speech Markup (SSML)

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="maincourse">
      <noinput> <reprompt/> </noinput>
      <nomatch> <reprompt/> </nomatch>
      <prompt count="1">Please select an entree.</prompt>
      <prompt count="2">Today, we're featuring:<enumerate/></prompt>
      <option dtmf="1" value="fish"> swordfish </option>
      <option dtmf="2" value="beef"> roast beef </option>
      <option dtmf="3" value="chicken"> frog legs </option>
      <filled>
        <prompt>
          <value expr="maincourse"/>, good choice.
          Please enjoy your meal.
        </prompt>
        <submit next="maincourse.cgi"
          method="post" namelist="maincourse"/>
      </filled>
    </field>
  </form>
</vxml>
```

<property>

Description

Control implementation platform settings. Properties are used to set values that affect platform behavior, such as the recognition process, timeouts, caching policy, etc.

Please refer to Properties for list of properties.

Syntax

```
<property  
  name="String"  
  value="String"/>
```

Attributes

Attribute	Description
name	Property name. <i>Required</i>
value	Property value. <i>Required</i>

Parents

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

Children

None.

VoiceGenie Extensions

Please refer to Properties for list of VoiceGenie properties.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <property name="bargain" value="true"  
  <form>  
    <field name="hello">  
      <grammar>hello</grammar>  
      <prompt>This prompt can be barged in. Please say hello</prompt>  
      <noinput>Say something</noinput>  
      <filled>  
        <prompt>Hello, world!</prompt>  
        <goto nextitem="helloagain"/>  
      </filled>  
    </field>  
    <field name="helloagain">  
      <property name="bargain" value="false"/>  
      <grammar>hello</grammar>  
      <prompt>This prompt cannot be barged in.  
        Please say hello</prompt>
```

```
<noinput>Say something</noinput>  
<filled>  
  <prompt>Hello, world!</prompt>  
</filled>  
</field>  
</form>  
</vxml>
```

<record>

Description

Collects a recording from the user. A reference to the recorded audio is stored in the field item variable, which can be played back or submitted to a server.

Syntax

```
<record
  name="String"
  expr="ECMAScript_Expression"
  cond="ECMAScript_Expression"
  modal="true" | "false"
  beep="true" | "false"
  maxtime="time_interval"
  mintime="time_interval"
  beginsilence="time_interval"
  finalsilence="time_interval"
  agc="on" | "off"
  dtmfterm="true" | "false"
  type="MIME_type"
  dest="URI"
  destexpr="ECMAScript_Expression">
  child elements
</record>
```

Attributes

Attribute	Description
name	Defines the field item variable that holds the recording. <i>Required</i>
expr	Initial value of the field item variable. <record> will not be executed if the value is not undefined. <i>Optional (Defaults to undefined)</i>
cond	Boolean expression that must evaluate to true in order for <record> to execute. <i>Optional (Defaults to true)</i>
modal	If this is true all higher-level speech and DTMF grammars are turned off while making the recording. If this is false, speech and DTMF grammars scoped to the form, document, application, and calling documents are listened for. <i>Optional (Defaults to true)</i>
beep	If true, a tone is emitted just prior to recording. <i>Optional (Defaults to false)</i>
maxtime	Maximum duration to record. Minimum value is 100ms. <i>Optional</i>
mintime	If the duration of the recording is less than this attribute, then it is assumed empty recording and throws noinput. Minimum value is 250ms. <i>Optional (Defaults to 250ms)</i>
beginsilence	The interval of silence before recording. <i>Optional (Defaults to 4s)</i>
finalsilence	The interval of silence that indicates end of speech. <i>Optional (Defaults to 4s)</i>
agc	Enables Auto Gain Control. <i>Optional (Defaults to on)</i>
dtmfterm	If true, a DTMF keypress terminates recording. The DTMF tone is not part of the recording. <i>Optional (Defaults to true)</i>
type	Media format of the resulting recording: <i>Optional (Defaults to audio/basic)</i> <ul style="list-style-type: none"> • audio/vox - same as audio/basic. • audio/wav • audio/adpcm • audio/adpcm8 • audio/adpcm6

dest	The URI for the destination of the recording. <i>Optional</i>
destexpr	Expression that yields the URI for destination of the recording. <i>Optional</i>

Shadow Variables

The following shadow variables are available in the same scope of the field item variable called name\$:

Shadow Variable	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 null.
name\$.maxtime	true if the recording was terminated because the maxtime duration was reached.
name\$.dest	URI referencing the recording. If the destination redirects the original request, for instance in the case of RTSP or HTTP servers, this variable holds the redirected URI.
name\$.filetype	Media format of the recorded audio file.

Parents

<form>

Children

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

VoiceGenie Extensions

- Added beginsilence, mintime, agc, destexpr attributes
- Added filetype shadow variable

Limitations/Restrictions

- <grammar> element within <record> is not supported. (Optional for VoiceXML)

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <record name="awaymessage" beep="true"
      maxtime="10s" finalsilence="4s" dtmfterm="true">
      <prompt>
        At the tone, please record your message
      </prompt>
    </record>
    <field name="confirm" type="boolean">
      <prompt>
        The message is <value expr="awaymessage"/>
      </prompt>
      <prompt>
        To keep it, please say yes. To discard it, say no
      </prompt>
      <filled>
        <if cond="comfirm">
          <submit next="saveaudio.pl" method="post">
```

```
        namelist="awaymessage" />  
    </if>  
  </filled>  
</field>  
</form>  
</vxml>
```

<reprompt>

Description

Replays previously played prompt. Normally the interpreter suppresses playing of prompts on the next form item after executing a <catch> element. However, if a <reprompt> is executed in the catch, the interpreter will normal prompt processing for the next form item (which includes selection of a prompt and incrementing the prompt counter).

Syntax

```
<reprompt />
```

Parents

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

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="hello">
      <grammar>hello</grammar>
      <prompt>Say hello</prompt>
      <noinput><reprompt /></noinput>
      <nomatch><reprompt /></nomatch>
      <filled>
        <prompt>Hello, world!</prompt>
      </filled>
    </field>
  </form>
</vxml>
```

<return>

Description

The <return> element completes execution of <subdialog> and returns control and data to a calling dialog.

Syntax

```
<return
  event="event "
  namelist="variable1 variable2 ..." />
```

Attributes

Attribute	Description
event	Return and throw this event. <i>Optional</i>
namelist	Space-separated list of variables to be returned to the calling dialog. <i>Optional</i> (Defaults to no variables)

Parents

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

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

- Value of variables in the namelist attributes will be converted to string type.

Example

Document that calls the subdialog:

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <subdialog name="result" src="subdialog.vxml">
      <filled>
        <prompt>Your account number is <value expr="result.acctnum"/>
          Your phone number is <value expr="result.acctphone"/>
        </prompt>
      </filled>
    </subdialog>
  </form>
</vxml>
```

Document containing the subdialog:

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form id="basic">
    <field name="acctnum" type="digits">
      <prompt> What is your account number? </prompt>
```

```
</field>  
<field name="acctphone" type="phone">  
  <prompt> What is your home telephone number? </prompt>  
  <filled>  
    <return namelist="acctnum acctphone"/>  
  </filled>  
</field>  
</form>  
</vxml>
```

<script>

Description

Includes a block of client-side script. Each <script> element is executed in the scope of its containing element; i.e., it does not have its own scope. Variables defined in <script> are equivalent to variables defined using <var> within the same scope.

Syntax

```
<script
  src="URI"
  charset="Encoding"
  fetchhint="prefetch" | "safe"
  fetchtimeout="time_interval"
  maxage="time_interval"
  maxstale="time_interval">
  Script Text
</script>
```

Attributes

Attribute	Description
src	The URI specifying the location of the external script. <i>Optional</i>
charset	Character encoding if external script is used. <i>Optional</i>
fetchtimeout	The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i>
fetchhint	Defines when the audio file should be retrieved. <i>Optional</i> <ul style="list-style-type: none">• prefetch - audio file may be downloaded when the page is loaded• safe - only load the audio file when needed
maxage	Indicates the maximum time in seconds that this document will use this script file before fetching another copy. <i>Optional</i>
maxstale	Indicates the maximum time in seconds that this document will use a script file whose exceeded the maxage time. <i>Optional</i>

Parents

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

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

It is wise to put CDATA escapes around your scripts so you don't have to escape XML reserved characters (eg. <, >, &, etc).

```
<?xml version="1.0"?>
```

```
<vxml version="2.0">
  <script>
    <![CDATA[
      function alwaysTrue() {
        // no need to escape the following less than sign
        return 1 < 2;
      }
    ]]>
  <block>
    <prompt>Hello</prompt>
    <prompt cond="alwaysTrue">I am always here</prompt>
  </block>
</vxml>
```

<subdialog>

Description

Invokes another dialog as a subdialog of the current one. The subdialog is a reusable dialog that allows values to be returned. The subdialog executes in a new execution context with all variables and execution state initialized. Values can be passed into the subdialog using <param> child elements; the subdialog must contain <var> variable declaration for each parameter. The original dialog can continue execution only when the subdialog executes <return> element. Returned values are available as properties of the <subdialog> field item variable.

Syntax

```
<subdialog
  name="String"
  expr="ECMAScript_Expression"
  cond="ECMAScript_Expression"
  namelist="variable1 variable2 ..."
  src="URI"
  srcexpr="ECMAScript_Expression"
  method="get" | "post"
  enctype="MIME_type"
  fetchaudio="URI"
  fetchhint="prefetch" | "safe"
  fetchtimeout="time_interval"
  maxage="time_interval"
  maxstale="time_interval">
  Script Text
</script>
```

Attributes

Attribute	Description
name	Name of field item variable for <subdialog>. The returned results can be retrieved as a property of the variable: name.returnVariable <i>Required</i>
expr	The initial value of the form item variable. <subdialog> will be visited only if the variable is undefined. <i>Optional (Defaults to undefined)</i>
cond	Boolean expression that must evaluate to true for <subdialog> to execute. <i>Optional (Defaults to true)</i>
namelist	Space-separated list of variables to submit. <i>Optional (Defaults to nothing)</i>
src	URI of the subdialog. <i>Required (src or srcexpr is required)</i>
srcexpr	Expression that yields the URI of the subdialog. <i>Required (src or srcexpr is required)</i>
method	Specifies the query request method: get or post <i>Optional (Defaults to get)</i>
enctype	MIME encoding of the document. The following types are supported: <i>Optional (Defaults to application/x-www-form-urlencoded)</i> <ul style="list-style-type: none"> • application/x-www-form-urlencoded • multipart/form-data
fetchaudio	The URI of audio to play while waiting for the next document to be fetched.
fetchtimeout	The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i>
fetchhint	Defines when the audio file should be retrieved. <i>Optional</i> <ul style="list-style-type: none"> • prefetch - audio file may be downloaded when the page is loaded • safe - only load the audio file when needed
maxage	Indicates the maximum time in seconds that this document will use this subdialog file

maxstale	before fetching another copy. <i>Optional</i> Indicates the maximum time in seconds that this document will use a subdialog file whose exceeded the maxage time. <i>Optional</i>
----------	---

Parents

<form>

Children

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

VoiceGenie Extensions

- added srcexpr attribute

Limitations/Restrictions

- Variables submitted to a subdialog in namelist attribute may not be objects.

Example

Document that calls the subdialog:

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <subdialog name="result" src="subdialog.vxml">
      <filled>
        <prompt>Your account number is <value expr="result.acctnum"/>
          Your phone number is <value expr="result.acctphone"/>
        </prompt>
      </filled>
    </subdialog>
  </form>
</vxml>
```

Document containing the subdialog:

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form id="basic">
    <field name="acctnum" type="digits">
      <prompt> What is your account number? </prompt>
    </field>
    <field name="acctphone" type="phone">
      <prompt> What is your home telephone number? </prompt>
      <filled>
        <return namelist="acctnum acctphone"/>
      </filled>
    </field>
  </form>
</vxml>
```

<submit>

Description

Submit values to document server. It is similar to <goto> in that it results in a new document being obtained. Unlike <goto>, it lets you submit a list of variables to the document server via HTTP GET or HTTP POST.

Syntax

```
<submit
  next="URI"
  expr="ECMAScript_Expression"
  namelist="variable1 variable2 ..."
  method="get" | "post"
  enctype="MIME_type"
  fetchaudio="URI"
  fetchaudiodelay="time_interval"
  fetchaudiominimum="time_interval"
  fetchhint="prefetch" | "safe"
  fetchtimeout="time_interval"
  maxage="time_interval"
  maxstale="time_interval">
  Script Text
</script>
```

Attributes

Attribute	Description
next	URI to which the query is submitted. <i>Required (next or expr is required)</i>
expr	Expression that yields the URI. <i>Required (next or expr is required)</i>
namelist	Space-separated list of variables to submit. <i>Optional (Defaults to nothing)</i>
method	Specifies the query request method: get or post <i>Optional (Defaults to get)</i>
enctype	MIME encoding of the document. The following types are supported: <i>Optional (Defaults to application/x-www-form-urlencoded)</i> <ul style="list-style-type: none"> • application/x-www-form-urlencoded • multipart/form-data
fetchaudio	The URI of audio to play while waiting for the next document to be fetched.
fetchaudiodelay	The time interval to wait before playing the fetchaudio source.
fetchaudiominimum	The minimum time interval to play the fetchaudio source, even if the fetch document arrives in the meantime.
fetchtimeout	The time interval to wait for audio file to be fetched before playing the alternate content. <i>Optional</i>
fetchhint	Defines when the audio file should be retrieved. <i>Optional</i> <ul style="list-style-type: none"> • prefetch - audio file may be downloaded when the page is loaded • safe - only load the audio file when needed
maxage	Indicates the maximum time in seconds that this document will use this file before fetching another copy. <i>Optional</i>
maxstale	Indicates the maximum time in seconds that this document will use a file whose exceeded the maxage time. <i>Optional</i>

Parents

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

Children

None.

VoiceGenie Extensions

- Attribute fetchaudiodelay is added.
- Attribute fetchaudiominimum is added.

Limitations/Restrictions

- Attribute fetchhint= prefetch is not supported (Optional in VoiceXML)

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
<form id="get_from_and_to_cities">
  <grammar src="from_to.grammar"/>
  <block>
    Welcome to the Driving Directions By Phone.
  </block>
  <initial name="bypass_init">
    <prompt>Where do you want to drive from and to?</prompt>
    <nomatch count="1">
      Please say something like this,
      "from Toronto Ontario to Ottawa Ontario".
    </nomatch>
    <nomatch count="2">
      I'm sorry, I still don't understand.
      I'll ask you for information one piece at a time.
      <assign name="bypass_init" expr="true"/>
      <reprompt/>
    </nomatch>
  </initial>
  <field name="from_city">
    <grammar src="city.grammar"/>
    <prompt>From which city are you leaving?</prompt>
    ...
  </field>
  <field name="to_city">
    <grammar src="city.grammar"/>
    <prompt>Which city are you going to?</prompt>
    ...
  </field>
  <block>
    <prompt>Retrieving directions</prompt>
    <submit next="direction.cgi" method="post"
      namelist="from_city to_city"/>
  </block>
</form>
</vxml>
```

<throw>

Description

Throws a pre-defined event or application-specific event.

Syntax

```
<throw
  event="event "
  eventexpr="ECMAScript_Expression"
  message="String"
  messageexpr="ECMAScript_Expression"/>
```

Attributes

Attribute	Description
event	Event name to throw. <i>Required (event or eventexpr is required)</i>
eventexpr	Expression that yields the event name. <i>Required (event or eventexpr is required)</i>
message	A message string providing additional context about the event being thrown. For the pre-defined events thrown by the platform, the value of the message is platform-dependent. The message will be available as a variable within the scope of the catch element. <i>Optional</i>
messageexpr	Expression that yields the message. <i>Optional</i>

Parents

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

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <catch event="goodbye">
    <prompt>goodbye.</prompt>
    <exit/>
  </catch>
  <form>
    <field name="hello">
      <grammar>hello | goodbye</grammar>
      <help>Just say hello</prompt>
      <prompt>Say hello</prompt>
      <noinput>Say something</noinput>
      <filled>
        <if cond="hello == 'goodbye'">
```

```
        <throw event="goodbye"/>
    </if>
    <prompt>Hello, world!</prompt>
</filled>
</field>
</form>
</vxml>
```

<transfer>

Description

Transfers the caller to another phone number. There are 2 types of transfer:

- bridge transfer - The caller resumes the interpreter session after the call with the third party completes
- blind transfer -As soon as the call with the third party connects, the platform throws a `telephone.disconnect.transfer` and terminates the interpreter session.

Syntax

```
<transfer
  name="String"
  expr="ECMAScript_Expression"
  cond="ECMAScript_Expression"
  dest="URI"
  destexpr="ECMAScript_Expression"
  bridge="true" | "false"
  connecttimeout="time_interval"
  maxtime="time_interval"
  transferaudio="URI"
  analysis="true" | "false"
  connectwhen="analysis" | "answered" | "immediate"
  uuidata="">
  child elements
</transfer>
```

Attributes

Attribute	Description
name	Name of field item variable. It stores the result of the <transfer>: <i>Required</i> <ul style="list-style-type: none">• busy - the endpoint refused the call.• noanswer - There was no answer within the time specified by the connecttimeout attribute.• network_busy - Some intermediate network refused the call.• near_end_disconnect - The call was completed and was terminated by the caller.• far_end_disconnect - The call was completed and was terminated by the callee.• network_disconnect - The call was completed and was terminated by the network.• maxtime_disconnect - The call duration exceeded the value of maxtime attribute and was terminated by the platform.• unknown - outcome of transfer is unknown.• not_allowed - not allowed to use <transfer> element for this document.• far_end_machine - detects answering machine picking up the call when analysis attribute is set to true.• far_end_fax - detects fax machine picking up the call when analysis attribute is set to true.• invalid_phone_no - phone number in dest or destexpr attribute is not a valid phone number format.

	<ul style="list-style-type: none"> restricted_phone_no - phone number is restricted.
expr	Expression that assigns the initial value of the field item variable. The <transfer> element will be executed only if the value is undefined. <i>Optional (Defaults to undefined)</i>
cond	Boolean expression that must evaluate to true for this <transfer> element to execute. <i>Optional (Defaults to true)</i>
dest	URI of the destination phone. Format is: phone : / / (416) 736-0905x111, where brackets and dashes are optional, and x represents extension number. <i>Required (dest or destexpr is required)</i>
destexpr	Expression that yields the destination. <i>Required (dest or destexpr is required)</i>
bridge	Determines what to do after call is connected. Please refer to description of this element for the 2 types of transfer. <i>Optional (Defaults to true)</i> <ul style="list-style-type: none"> true - bridge transfer false - blind transfer
connecttimeout	The time to wait while trying to connect the call before returning the noanswer condition. Minimum value allowed is 5s. <i>Optional (Defaults to 30s)</i>
maxtime	The maximum time that the call is allowed to last. Minimum value is 30 seconds; maximum is one week. 0 for no limit. This attribute only applies when bridge=true. <i>Optional (Defaults to 0)</i>
transferaudio	URI of audio file to be play while connecting the call. <i>Optional</i>
analysis	Set to true to enable detection of whom answered the call. <i>Optional (Defaults to false)</i>
connectwhen	Controls when the call is connected to end point. <i>Optional (Defaults to immediate)</i> <ul style="list-style-type: none"> analysis - connection is made after call is connected and analysis is completed (only applies when analysis=true) answered - connection is made after call is picked up immediate - user hears the ringing
uuidata	Sends User-to-User Information (UII) to the outbound call. <i>Optional (Sends uuidata from inbound line)</i>

Shadow Variables

The <transfer> shadow variable (name\$) has the following properties after the transfer completes:

- name\$duration - the duration of a successful call in seconds. If duration is 0, the call was terminated before answered.

Events Thrown

The following events can be thrown during the execution of <transfer> element:

Event	Description
telephone.disconnect.hangup	Caller hangs up.
telephone.disconnect.transfer	Call was "blind transferred" to another line and will not return.
error.telephone.noauthorization	Caller is not allowed to call the destination.
error.telephone.baddestination	Destination URI is malformed.
error.telephone.noroute	The platform is not able to place a call to the destination.
error.telephone.noresource	The platform cannot allocate resource to place the call.

Parents

<form>

Children

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

VoiceGenie Extensions

- Added return values for field item variables: not_allowed, far_end_machine, far_end_fax, invalid_phone_no, restricted_phone_no.
- Added attributes analysis, connectwhen, uuidata.

Limitations/Restrictions

- <grammar> element within <transfer> is not supported. (Optional for VoiceXML)
- Only <dtmf> elements within a <transfer> can be used to specify DTMF grammar that will terminate the transfer. DTMF grammars outside of the scope of <transfer> are not enabled.
-

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
<form>
  <var name="calldur" expr="0"/>
  <block>
    <prompt>
      Welcome to VoiceGenie
    </prompt>
  </block>
  <transfer name="newcall" dest="phone://(416)7360905x111"
    connecttimeout="10s" bridge="true">
    <filled>
      Your call lasted <value expr="newcall$.duration"/> seconds.
      <if cond="newcall == 'busy'">
        <prompt>
          All our customer care agents are currently busy.
          Please call back later.
        </prompt>
      </if>
    </filled>
  </transfer>
</form>
</vxml>
```

<value>

Insert the value of an expression in a prompt.

Syntax

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

Attributes

Attribute	Description
expr	Expression to return. <i>Required</i>

Parents

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

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

- The value evaluated in expr attribute may not contain Speech Markup fragments.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <form>  
    <field name="maincourse">  
      <noinput> <reprompt/> </noinput>  
      <nomatch> <reprompt/> </nomatch>  
      <prompt count="1">Please select an entree.</prompt>  
      <prompt count="2">Today, we're featuring:<enumerate/></prompt>  
      <option dtmf="1" value="fish"> swordfish </option>  
      <option dtmf="2" value="beef"> roast beef </option>  
      <option dtmf="3" value="chicken"> frog legs </option>  
      <filled>  
        <prompt>  
          <value expr="maincourse"/>, good choice.  
          Please enjoy your meal.  
        </prompt>  
        <submit next="maincourse.cgi"  
          method="post" namelist="maincourse"/>  
      </filled>  
    </field>  
  </form>  
</vxml>
```

<var>

Description

Declares a variable in the scope of its parent element. If the variable is already defined in the current scope, further declaration is treated as value assignments.

Syntax

```
<var  
  name="String"  
  expr="ECMAScript_Expression"/>
```

Attributes

Attribute	Description
name	Name of variable. <i>Required</i>
expr	Initial value of variable. <i>Optional (Defaults to undefined)</i>

Parents

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

Children

None.

VoiceGenie Extensions

None.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>  
<vxml version="2.0">  
  <var name="world" expr="'world'"/>  
  <form>  
    <field name="hello">  
      <grammar>hello | goodbye</grammar>  
      <help>Just say hello</prompt>  
      <prompt>Say hello</prompt>  
      <noinput>Say something</noinput>  
      <filled>  
        <prompt>Hello, <value expr="world"/>!</prompt>  
      </filled>  
    </field>  
  </form>  
</vxml>
```

<vxml>

Description

Identifies a VoiceXML document.

Syntax

```
<vxml
  version="Version"
  base="URI"
  xml:lang="Language"
  application="URI"
  logpage="1" />
```

Attributes

Attribute	Description
version	Specifies the VoiceXML version number. The current version is 2.0. <i>Required</i>
base	The base URI of this document. Relative references in this document use this URI as the base URI. <i>Optional</i>
xml:lang	The language and locale type for this document. <i>Optional (Defaults to en-US)</i>
application	The URI of this document's application root document. If the root document cannot be found or the root document has reference to another root document, an <code>error.semantic</code> event is thrown. <i>Optional</i>
logpage	If value is 1, sends logging information to maintainer (can be set using <code><meta></code> element) <i>Optional</i>

Parents

None.

Children

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

VoiceGenie Extensions

- Added logpage attribute.

Limitations/Restrictions

None.

Example

```
<?xml version="1.0"?>
<vxml version="2.0">
  <form>
    <field name="hello">
      <grammar>hello | goodbye</grammar>
      <help>Just say hello</prompt>
      <prompt>Say hello</prompt>
      <noinput>Say something</noinput>
      <filled>
        <prompt>Hello, world!</prompt>
```

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

Speech Markup Reference

VoiceGenie VoiceXML now uses the Speech Synthesis Markup Language (SSML) to control TTS pronunciation, rate, pitch, and other characteristics. You can include speech markup tags within <prompt>, <audio>, <choice>, and <enumerate> elements.

VoiceXML 1.0 uses Java API Speech Markup Language (JSML) for speech markups, but JSML elements have been replaced by SSML in VoiceXML 2.0. Therefore, the JSML elements <emp>, <div>, <pros>, and <sayas> are now obsolete in VoiceXML 2.0.

VoiceGenie Limitation

- Speech markup is not supported within the <choice> element
- Alternate text within <audio> element only supports one element (either a string, a <value> element, an <audio> element, or a speech markup element)

<break>

Description

Specifies a pause in the speech output.

Syntax

```
<break  
  time="time_interval"  
  size="none" | "small" | "medium" | "large" />
```

Attributes

Attribute	Description
time	Specifies an absolute pause in seconds or milliseconds. <i>Optional (Only one of time and expr can be defined)</i>
size	Specifies relative pause duration. The actual pause duration depends on TTS engine. <i>Optional (Defaults to medium)</i>

Parents

<emphasis>, <paragraph>, <prosody>, <sentence>, <voice>

Children

None.

Example

To include speech markup tags, insert the SSML tags inside one of the <prompt>, <audio>, <choice>, and <enumerate> elements.

```
<prompt>  
  Hello, <break size="large"> world!  
</prompt>
```

<emphasis>

Description

Specifies that the enclosed text should be spoken with emphasis

Syntax

```
<emphasis  
  level="strong" | "moderate" | "none" | "reduced">  
  child elements  
</emphasis >
```

Attributes

Attribute	Description
level	Specifies the level of emphasis. <i>Optional (Defaults to moderate)</i>

Parents

<paragraph>, <sentence>

Children

<break>, <emphasis>, <phoneme>, <prosody>, <say-as>, <voice>

Limitation

<emphasis> element is not supported by Speechify.

Example

To include speech markup tags, insert the SSML tags inside one of the <prompt>, <audio>, <choice>, and <enumerate> elements.

```
<prompt>  
  Hello, <emphasis level="strong">world!</emphasis>  
</prompt>
```

<paragraph>

Description

Identifies the enclosed text as a paragraph, containing zero or more sentences. For brevity, <p> is equivalent to <paragraph>.

Syntax

```
<paragraph  
  xml:lang="Locale">  
  child elements  
</paragraph>
```

Attributes

Attribute	Description
xml:lang	Defines the language and locale for this paragraph. <i>Optional</i>

Parents

None.

Children

<break>, <emphasis>, <phoneme>, <prosody>, <say-as>, <sentence>, <voice>

Example

To include speech markup tags, insert the SSML tags inside one of the <prompt>, <audio>, <choice>, and <enumerate> elements.

```
<prompt>
  <paragraph>
    <sentence>This is the first sentence of first paragraph.</sentence>
    <sentence>second sentence of first paragraph.</sentence>
  </paragraph>
</prompt>
```

<phoneme>

Description

Specifies a phonetic pronunciation for the contained text.

Syntax

```
<phoneme
  ph="String"
  alphabet="ipa" | "worldbet" | "xsampa">
  text
</phoneme>
```

Attributes

Attribute	Description
ph	Specifies the phonetic string. <i>Required</i>
alphabet	Specifies the alphabet to use. <i>Optional</i>

Parents

<emphasis>, <paragraph>, <prosody>, <sentence>, <voice>

Children

None.

<prosody>

Description

Specifies prosodic information for the enclosed text.

Syntax

```
<prosody
```

```
pitch="high" | "medium" | "low" | "default"  
contour="contour"  
range="high" | "medium" | "low" | "default"  
rate="fast" | "medium" | "slow" | "default"  
duration="time_interval"  
volume="silent" | "soft" | "medium" | "loud" | "default">  
  child elements  
</prosody>
```

Attributes

Attribute	Description
pitch	Specifies the pitch.
contour	Sets the actual pitch contour for the text.
range	Specifies pitch range.
rate	Specifies speaking rate.
duration	Specifies the duration in seconds or milliseconds for the time to read the text.
volume	Specifies output volume.

Parents

<emphasis>, <paragraph>, <prosody>, <sentence>, <voice>

Children

<break>, <emphasis>, <paragraph>, <phoneme>, <prosody>, <say-as>, <sentence>, <voice>

Limitation

<prosody> element is not supported by Speechify.

Example

To include speech markup tags, insert the SSML tags inside one of the <prompt>, <audio>, <choice>, and <enumerate> elements.

```
<prompt>  
  <prosody pitch="high" duration="2s">Ouch!</prosody>  
</prompt>
```

<say-as>

Description

Specifies the type of text construct contained within the element.

Syntax

```
<say-as  
  sub="String"  
  type="construct_type">  
  text  
</say-as>
```

Attributes

Attribute	Description
sub	Defines substitute text to be spoken instead of the contained text.

type	Indicates the text construct type. The values of type can optionally contain a colon and format: <ul style="list-style-type: none">• acronym - pronounce text as individual characters• number - contained text is a number. Format values are ordinal and digits• date - contained text is a date. Date formats are a combination of year "y", month "m", and date "d".• time - contained text is a time of day. Format values are hm or hms.• duration - contained text is a time duration. Format values are hm, hms, or ms• currency - contained text is current amount.• measure - contained text is a measurement.• telephone - contained text is a telephone number.• name - contained text is a name of person.• net - contained text is an internet handle. Format values are email or url.• address - contained text is a postal address.
------	---

Parents

<emphasis>, <paragraph>, <prosody>, <sentence>, <voice>

Children

None.

Example

To include speech markup tags, insert the SSML tags inside one of the <prompt>, <audio>, <choice>, and <enumerate> elements.

```
<prompt>
  <sayas type="acronym">XML</sayas>
  Today's date is <sayas type="date:ymd">2001/01/01</sayas>
  You still owe me <sayas type="currency">$12345.67</sayas>
  My email address is
    <sayas type="net:email">email@voicegenie.com</sayas>
  I work for <sayas sub="Voice Genie">VG</sayas>
</prompt>
```

<sentence>

Description

Identifies the enclosed text as a sentence, containing zero or more sentences. For brevity, <s> is equivalent to <sentence>.

Syntax

```
<sentence
  xml:lang="Locale">
  child elements
</sentence >
```

Attributes

Attribute	Description
xml:lang	Defines the language and locale for this sentence. <i>Optional</i>

Parents

None.

Children

<break>, <emphasis>, <phoneme>, <prosody>, <say-as>, <voice>

Example

To include speech markup tags, insert the SSML tags inside one of the <prompt>, <audio>, <choice>, and <enumerate> elements.

```
<prompt>
  <paragraph>
    <sentence>This is the first sentence of first paragraph.</sentence>
    <sentence>second sentence of first paragraph.</sentence>
  </paragraph>
</prompt>
```

<voice>

Description

Specifies voice characteristics for the spoken text.

Syntax

```
<voice
  gender="male" | "female" | "neutral"
  age="integer"
  category="child" | "teenager" | "adult" | "elder"
  variant="integer"
  name="name" >
  child elements
</prosody>
```

Attributes

Attribute	Description
gender	Preferred gender of the voice to speak the text.
age	Preferred age of the voice to speak the text.
category	Preferred age category of the voice to speak the text.
variant	A variant of the other voice characteristics to speak the text.
name	Name of the specific voice name to speak the text.

Parents

<emphasis>, <paragraph>, <prosody>, <sentence>, <voice>

Children

<break>, <emphasis>, <paragraph>, <phoneme>, <prosody>, <say-as>, <sentence>, <voice>

Limitation

<voice> element is not supported by Speechify.

Example

To include speech markup tags, insert the SSML tags inside one of the <prompt>, <audio>, <choice>, and <enumerate> elements.

```
<prompt>  
  <voice gender="female">I am a woman</voice>  
  <voice gender="elder">I am getting old</voice>  
  <!-- platform specific voice name -->  
  <voice name="mike">My name is Mike</voice>  
</prompt>
```

ECMAScript Reference

ECMAScripts is the client-side scripting language for VoiceXML, and is analogous to JavaScript in HTML. Scripts can be included within the <script> element as a block of statements. There are also many attributes in VoiceXML elements that accept script expressions as the attribute value (those attributes have ECMAScript_Expression type, usually called expr). ECMAScripts share the same variables and variable scope with the VoiceXML document, so declaring variables using <script> or <var> element will have the same effect.

All of the ECMAScript operators, statements, and core objects are supported by the VoiceGenie platform.

Reference

Netscape JavaScript Reference:

<http://developer.netscape.com/docs/manuals/js/core/jsref15/contents.html>

Netscape JavaScript Guide:

<http://developer.netscape.com/docs/manuals/js/core/jsguide15/contents.html>

Escape Characters in String Literals

Character	Description
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Tab
\v	Vertical Tab
'	Single quote
"	Double quote
\	Backslash
\???	The character with the Latin-1 encoding specified by up to three octal digits ??? between 0 and 377. For example, \251 is the octal sequence for the copyright symbol.
\x??	The character with the Latin-1 encoding specified by the two hexadecimal digits ?? between 00 and FF. For example, \xA9 is the hexadecimal sequence for the copyright symbol.
\u????	The Unicode character specified by the four hexadecimal digits ?????. For example, \u00A9 is the Unicode sequence for the copyright symbol.

Operations

Category	Operator	Description
Arithmetic	+	Adds 2 numbers.
	++	Increments a number.
	-	As a unary operator, negates the value of its argument. As a binary operator, subtracts 2 numbers.
	--	Decrements a number.
	*	Multiplies 2 numbers.
	/	Divides 2 numbers.
	%	Computes the integer remainder of dividing 2 numbers.
String	+	Concatenates 2 strings.
	+=	Concatenates 2 strings and assigns the result to the first operand.
Logical Operators	&&	(Logical AND) Returns true if both logical operands are true. Otherwise, returns false.

		(Logical OR) Returns true if either logical expression is true. If both are false, returns false.
	!	(Logical negation) If its single operand is true, returns false; otherwise, returns true.
Bitwise Operators	&	(Bitwise AND) Returns a one in each bit position if bits of both operands are ones.
	^	(Bitwise XOR) Returns a one in a bit position if bits of one but not both operands are one.
		(Bitwise OR) Returns a one in a bit if bits of either operand is one.
	~	(Bitwise NOT) Flips the bits of its operand.
	<<	(Left shift) Shifts its first operand in binary representation the number of bits to the left specified in the second operand, shifting in zeros from the right.
	>>	(Sign-propagating right shift) Shifts the first operand in binary representation the number of bits to the right specified in the second operand, discarding bits shifted off.
	>>>	(Zero-fill right shift) Shifts the first operand in binary representation the number of bits to the right specified in the second operand, discarding bits shifted off, and shifting in zeros from the left.
	Assignment	=
+=		Adds 2 numbers and assigns the result to the first.
-=		Subtracts 2 numbers and assigns the result to the first.
*=		Multiplies 2 numbers and assigns the result to the first.
/=		Divides 2 numbers and assigns the result to the first.
%=		Computes the modulus of 2 numbers and assigns the result to the first.
&=		Performs a bitwise AND and assigns the result to the first operand.
^=		Performs a bitwise XOR and assigns the result to the first operand.
=		Performs a bitwise OR and assigns the result to the first operand.
<<=		Performs a left shift and assigns the result to the first operand.
>>=	Performs a sign-propagating right shift and assigns the result to the first operand.	
>>>=	Performs a zero-fill right shift and assigns the result to the first operand.	
Comparison	==	Returns true if the operands are equal.
	!=	Returns true if the operands are not equal.
	>	Returns true if left operand is greater than right operand.
	>=	Returns true if left operand is greater than or equal to right operand.
	<	Returns true if left operand is less than right operand.
	<=	Returns true if left operand is less than or equal to right operand.
Special	?:	Performs simple "if ? then : else"
	,	Evaluates two expressions and returns the result of the second expression.
	delete	Deletes an object property or an element at a specified index in an array.
	new	Creates an instance of an object.
	this	Refers to the current object.
	typeof	Returns a string indicating the type of the unevaluated operand.
void	Specifies an expression to be evaluated without returning a value.	

Statements

Statement	Description
break	Terminates the current while or for loop and transfers program control to the statement following the terminated loop.
continue	Terminates execution of the block of statements in a while or for loop, and continues execution of the loop with the next iteration.
delete	Deletes an object's property or an element of an array.
do	Executes its statements until the test condition evaluates to false.
...	Statement is executed at least once.

<code>while(condition)</code>	
<code>for (init; condition; increment)</code>	A loop that consists of three optional expressions, enclosed in parentheses and separated by semicolons, followed by a block of statements executed in the loop.
<code>for (var in object)</code>	Iterates a specified variable over all the properties of an object. For each distinct property, JavaScript executes the specified statements.
<code>function</code>	Declares a JavaScript function name with the specified parameters. Acceptable parameters include strings, numbers, and objects.
<code>if (condition) ... else</code>	Executes a set of statements if a specified condition is true. If the condition is false, another set of statements can be executed.
<code>labeled</code>	Provides an identifier that can be used with <code>break</code> or <code>continue</code> to indicate where the program should continue execution.
<code>return</code>	Statement that specifies the value to be returned by a function.
<code>switch (expression)</code> <code>case label:</code>	Evaluates an expression and attempt to match the expression's value to a case label.
<code>var</code>	Declares a variable, optionally initializing it to a value.
<code>while (condition) ...</code>	Creates a loop that evaluates an expression, and if it is true, executes a block of statements.
<code>with (object) ...</code>	Establishes the default object for a set of statements.
<code>//</code>	Defines comment until end of the line.
<code>/* ... */</code>	Defines comment within the operators.

Core Objects

Array Object

Property	Description
<code>length</code>	Size of the array.
<code>index</code>	Position of matched substring (from <code>RegExp</code> object)
<code>input</code>	Original string for matching (from <code>RegExp</code> object)
Method	Description
<code>concat(array1)</code>	Joins two arrays into one array.
<code>join(separator)</code>	Joins array element into a string, separated by separator (Defaults to ",")
<code>pop</code>	Removes last element from an array and returns that element.
<code>push(e1, e2 ...)</code>	Adds one or more elements to the end of the array and returns the last element.
<code>reverse</code>	Reverses the elements in the array.
<code>shift</code>	Removes the first element from an array and returns that element.
<code>slice(begin, end)</code>	Extracts elements from index begin to end and returns a new array.
<code>sort</code>	Sorts the elements of an array
<code>splice</code>	Change content of array by adding and removing elements.
<code>toString</code>	Returns string representation of array
<code>unshift(e1, e2 ...)</code>	Adds one or more elements to the beginning of the array and returns new array length.

Boolean Object

Method	Description
<code>toString</code>	Returns string representation of Boolean.

Date Object

Method	Description
<code>getDate</code>	Returns the day of the month.
<code>getDay</code>	Returns the day of the week.
<code>getHours</code>	Returns the hour.
<code>getMinutes</code>	Returns the minutes.
<code>getMonth</code>	Returns the month.

getSeconds	Returns the seconds.
getTime	Returns the numeric value corresponding to the time.
getTimezoneOffset	Returns the time -zone offset in minutes for the current locale.
getYear	Returns the year.
parse	Returns the number of milliseconds in a date string since January 1, 1970, 00:00:00, local time.
setDate	Sets the day of the month.
setHours	Sets the hours.
setMinutes	Sets the minutes.
setMonth	Sets the month.
setSeconds	Sets the seconds.
setTime	Sets the value of a Date object.
setYear	Sets the year.
toGMTString	Converts a date to a string, using the Internet GMT conventions.
toLocaleString	Converts a date to a string, using the current locale's conventions.
UTC	Returns the number of milliseconds in a Date object since January 1, 1970, 00:00:00, Universal Coordinated Time (GMT).

Math Object

Property	Description
E	Euler's constant, approximately 2.718.
LN10	Natural logarithm of 10, approximately 2.302.
LN2	Natural logarithm of 2, approximately 0.693.
LOG10E	Base 10 logarithm of E, approximately 0.434.
PI	Pi, approximately 3.14159.
SQRT1_2	Square root of 1/2, approximately 0.707.
SQRT2	Square root of 2, approximately 1.414.
Method	Description
abs	Returns the absolute value of a number.
acos	Returns the arccosine (in radians) of a number.
asin	Returns the arcsine (in radians) of a number.
atan	Returns the arctangent (in radians) of a number.
atan2	Returns the arctangent of the quotient of its arguments.
ceil	Returns the smallest integer greater than or equal to a number.
cos	Returns the cosine of a number.
exp	Returns E^{number} , where number is the argument, and E is Euler's constant, the base of the natural logarithms.
floor	Returns the largest integer less than or equal to a number.
log	Returns the natural logarithm (base E) of a number.
max	Returns the greater of two numbers.
min	Returns the lesser of two numbers.
pow	Returns base to the exponent power, that is, $\text{base}^{\text{exponent}}$.
random	Returns a pseudo-random number between 0 and 1.
round	Returns the value of a number rounded to the nearest integer.
sin	Returns the sine of a number.
sqrt	Returns the square root of a number.
tan	Returns the tangent of a number.

Number Object

Property	Description
MAX_VALUE	The largest representable number.

MIN_VALUE	The smaller representable number.
NaN	Not a number value.
NEGATIVE_INFINITY	Negative infinite value for overflow.
POSITIVE_INFINITY	Infinite value for overflow.
Method	Description
toString	Returns string representation of a number.

Object Object

Method	Description
eval	Evaluates a string of ECMAScript in the context of this object.
toString	Returns the string representation of this object.
valueOf	Returns the primitive value of the specified object.

String Object

Property	Description
length	Returns length of the string.
Method	Description
charAt	Returns the character at the specified index.
charCodeAt	Returns a number indicating the ISO-Latin-1 codeset value of the character at the given index.
concat	Combines the text of two strings and returns a new string.
fromCharCode	Returns a string from the specified sequence of numbers that are ISO-Latin-1 codeset values.
indexOf	Returns the index within the calling <code>String</code> object of the first occurrence of the specified value.
lastIndexOf	Returns the index within the calling <code>String</code> object of the last occurrence of the specified value.
match	Matches a regular expression against a string.
replace	Finds a match between a regular expression and a string, and to replace the matched substring with a new substring.
search	Executes the search for a match between a regular expression and a specified string.
slice	Extracts a section of a string and returns a new string.
split	Splits a string into an array of strings by separating the string into substrings.
substr	Returns the characters in a string beginning at the specified location through the specified number of characters.
substring	Returns the characters in a string between two indexes into the string.
toLowerCase	Returns the calling string value converted to lowercase.
toUpperCase	Returns the calling string value converted to uppercase.

RegExp Object

Property	Description
\$1...\$9	Parenthesized substring matches, if any.
global	Whether or not to test the regular expression against all possible matches in a string, or only against the first.
ignoreCase	Whether or not to ignore case while attempting a match in a string.
input or \$ <u>_</u>	The string against which a regular expression is matched.
lastIndex	The index at which to start the next match.
lastMatch or \$ <u>&</u>	The last matched characters.
lastParen	The last parenthesized substring match, if any.
leftContext or \$ <u>`</u>	The substring preceding the most recent match.
multiline or \$ <u>*</u>	Whether or not to search in strings across multiple lines.

right Context or \$'	The substring following the most recent match.
source	The text of the pattern.
Method	Description
compile	Compiles a regular expression object.
exec	Executes a search for a match in its string parameter.
test	Tests for a match in its string parameter.