



## VoiceGenie VoiceXML Gateway 4.6R: VoiceXML Support Details

### Overview

This release of the VoiceGenie VoiceXML Gateway software includes a number of changes, enhancements and bug fixes. Details of changes and problem resolution are provided in the “Change History” section. Usage notes are provided in the section “Usage Notes”. Finally, a complete tag reference summarizing support details is provided at the end of this document.

### Summary of New Features in this Release

This release of the VoiceGenie interpreter includes a number of features that have been requested by customers:

- Queued prompts (i.e. we now support barge-in for prompts queued in executable content);
- N-best for Nuance;
- SpeechWorks OSR 1.0 and Speechify 1.1 support;
- Improved GenieTracer support;
- A number of important bug fixes;

In addition, our support for the SpeechGenie platform (watch for it!) and VoiceXML 2.0 is even better.

### Change History

#### Changes from Release 4.5 to Release 4.6

- Only add "#" to URL when target is specified;
- Changes to support queued prompts for executable content, including barge-in;
- More support for GenieTracer (if you haven't tried it, you should!);
- N-best support for Nuance;
- Better support for 'file:' scheme in <dtmf>, <grammar>, <audio>



- Support for SSML with `<value mode=tts class=...>`
- Support `<say-as>` (and) `<sayas>` for SSML support;
- For `class=date` or `time` in `<value mode=tts...>`, use `type=date:yymd` or `type=time:hm` with `<say-as>` tag in SSML;
- Added attribute `UIDATA` to `<transfer>`

### Changes from Release 2.4 to Release 4.5

- Support for cookies;
- Added `<log>` tag;
- Added support for `EXPR` attribute in `<grammar>`, `<dtmf>` and `<audio>`;
- Added support for `fetchaudio` with `<subdialog>`;
- Added attributes `ANALYSIS` and `CONNECTWHEN` to `<transfer>`;
- Throw `error.unsupported.format` event for unknown ASR/DTMF grammar types;
- Remove fragment following '#' from URL during Web fetch;
- Handle grammar rule name specified after '#';
- With `<field type=...>`, support post processing of ASR or DTMF result with a predefined ECMAScript if it exists, for VoiceXML conformance etc.; DTMF output is processed for conformance even without the ECMAScript file;
- Added support for more literals in `<value mode=recorded...>`, and handle '?' in date string for `class=date`;
- Throw `error.asr` instead of `error.application` on ASR error;
- Added support for ASRENGINE MSSR, incorporated changes done for NT port;
- Added support for ASRENGINE SPEECHWORKS and LSS;
- Added support for inline XML grammar for SPEECHWORKS;
- Added support for n-best with property `MAXNBEST`, and field shadow variables `<name>$.nbestresult` and `<name>$.nbestconfidence`. This is currently supported only for SpeechWorks and Nuance ASR;
- Subject line in Maintainer e-mail is now 'VoiceGenie VoiceXML Log'; still from `PW_TRAP_IP`;



## Bug Fixes in Release 4.6

- Fixed a problem with playing certain types of WAVE files;
- Fixed some bugs related to EXPR support in <goto> and <submit>;
- Fixed a problem where the last prompt timeout was used for input collection;
- Fix for <goto nextitem>; we now always select and queue the prompt even if the nextitem has been visited already;

## Bug Fixes in Release 4.5

- Corrected a few issues with outbound calling;
- Tested with more complete range of Dialogic cards;
- Fixes related to answering machine detection;
- Better behavior if a TTS server fails;
- Fixed a bug related to DTMF input processing with TERMTIMEOUT > 0;
- Fixed some caching related bugs and changed behavior so safe caching is strictly observed even on same page;
- With <field type=...>, allow DTMF or ASR built in grammar to be active if <grammar> or <dtmf> is inside <field>, respectively;
- Fixed bug in form submit: if no fetchaudio to play, do not start audio;
- Fixed bugs related to fetchaudiotime (builtin, submit) and fetchaudiodelay;
- Fix a bug with <enumerate> when inputmodes=dtmf.;
- Fixed a bug where <return> in a subdialog did not have namelist or event.

## Changes from Release 2.3 to Release 2.4

- Support for Speechify TTS and SpeechWorks ASR (internal release only);



### **Changes from Release 2.2 to Release 2.3**

- A number of logging enhancements are now in place;
- Some issues regarding ASR control and performance have been addressed;
- DTMF bargein behavior has been normalized;
- We have added audio control support (see “Usage Notes”);
- There have been some improvements and extensions to our release of VoiceXML;
- There have been some improvements to the platform implementation;
- A number of enhancements supporting future releases have been added;

### **Bug Fixes in Release 2.3**

- Handle errors during <value> tag processing;
- Handle null URI properly;
- Handle HTTP error for VoiceXML pages properly;
- Fixed bug in reading HTTP header from a fetched ECMAScript file;
- Fixed problem related to subdialog - if same audio used in root document and the page that uses the root, it would kill vxmli;
- Fixed event in <link> - use context from where link is invoked instead of where it is defined;
- Check and throw error if grammar type is unsupported;
- Fixed bug with MSECS attribute value in <break>, and RANGE attribute in <pros>;
- Use TTSENGINE to add speech markup reset tags as necessary.

### **Changes from Release 2.1 to Release 2.2**

- Support for Speech markup;
- Default event handlers improved;



- Improved grammar support for <choice> and <menu>;

## Bug Fixes in Release 2.2

- The parser now supports the entity 'apos';
- Handle small input timeout better;
- If first element is not a <vxml>, throw “error.semantic”, instead of simply hanging up;
- ABNF grammar problem with <dtmf>; when there is an ambiguous match (e.g., 1 | 11) it will recognize only the shorter one - the workaround is to set TERMTIMEOUT property to 2s: this has been fixed;
- <choice> entries with no event/expr/next attribute are now handled properly;
- Handle null URI properly;
- Fixed <reprompt> with <menu>;
- Handle illegal/infinite loop in <form>;
- Bugfix related to subdialog - if same audio used in root doc and the page that uses the root, it would kill vxmli;
- Fixed event in <link> - use context from where link is invoked instead of where it is defined;
- Fixed bug in reading http header from a fetched ECMAScript file;
- Fixed bug in <record>: set field item variable on hangup so it could still be submitted in `telephone.disconnect.hangup` event;
- Fixed bug with MSECs attribute value in <break>.

## Usage Notes



## Speech Recognition Grammars

Mixed-initiative, <link> and global grammars are supported. All these need ASR or DTMF grammars that have slots associated with the return values. This means the ASR engine must support grammars with slots. The Nuance and SpeechWorks engines support this, but Watson does not. Note: when multiple slots are returned by a grammar and the grammar does not match any of the global grammars, the field variable gets the value of the slots and their values in the format of “+slot1:value1+slot2:value2...”. If only one slot is returned by the grammar, the field variable gets only the value of the slot, no matter what the slot’s name.

The new attribute `EXPR` can be used as an alternative for `SRC`. In this case, the ECMAScript expression is evaluated, and the result specifies the URI from which to fetch the grammar. Grammars may be specified in a number of different formats. The valid values for the `TYPE` attribute of the <grammar> tag are:

<code>application/x-abnf</code>	(for Nuance ASR)
<code>application/x-voicegenie-nuance</code>	(for Nuance ASR)
<code>application/x-voicegenie-watson</code>	(for Watson ASR)
<code>application/grammar-xml</code>	(for SpeechWorks ASR)
<code>application/x-swi-grammar-compiled</code>	(for SpeechWorks ASR)
<code>application/x-jsgf</code>	(for Lucent Speech Solutions (LSS) ASR)
<code>application/x-ms-xml</code>	(for Microsoft ASR)

If the type is ABNF, it will be converted to the ASR engine grammar format. This is supported with Nuance engine only. The default type of an inline grammar is ABNF for Nuance. The old simple ABNF format, `phrase1 | phrase2 ... | phrasen`, is supported for inline grammars with Watson and MSSR ASR. The default for external grammars is of the engine type, except for SpeechWorks and LSS. For SpeechWorks the default type for inline and external grammars is XML. For LSS, the default type for inline and external grammars is JSGF. With Nuance, slot names are required in grammars for choice, link and form/document/application level grammars.

If a type attribute is not specified, the **Content-type** header returned by the Web server is respected. If a type attribute is specified, it overrides any Content-type provided by the Web server. If the specified type or content-type are engine-specific, and do not match the `ASRENGINE` property, an error will be indicated. If the type is unknown, the default type will be ABNF for in-line grammars (unless the `ASRENGINE` is Watson, in which case only a simple disjunctive in-line grammar is supported). For external grammars



specified using the SRC attribute, the default depends upon the ASRENGINE property.

You can put complete Nuance, Watson, or ABNF grammars in-line using an XML CDATA block. The ASR engine grammar type can be used within CDATA for inline grammars.

For ABNF grammars (both DTMF and ASR), you can specify a slot for a return tag by appending the slot name with a ':' separator to the tag name: for example:

```
pizza {slot1:pizza} pop {slot2:pop}
```

This is necessary for mixed-initiative and global grammars, and may be useful for situations where multiple tags are returned, but you only want a specific tag for the field variable. ABNF grammars can also have weights attached. These are mapped to the appropriate underlying engines weights where supported.

The predefined VoiceXML types for currency, phone, number, digit, time and date require the appropriate grammar files under the grammar/<engine> sub directory for Nuance, Watson and MSSR (the file name should be <builtin name>.txt). When using builtin grammars with the TYPE attribute, the DTMF input or the ASR grammar results are processed by an ECMAScript script under the script/builtin directory to make it VoiceXML compliant before assigning the value to the field variable. For DTMF, the file is dtmf\_<builtin name>.js; for ASR, the file is <engine>\_<builtin name>.js, and if it does not exist and a dummy/empty file <engine>\_<builtin name>.na also does not exist, the default\_<builtin name>.js file may be used if it exists. **The dummy .na file is used to disable the post-processing if the "default\_\*" file exists.** Note that this is transparent in general use, but provides a method for post-processing the return values from pre-existing grammars so as to meet the VoiceXML specification.

**The prototype for each ECMAScript function is: for DTMF, "function parseDtmf(dtmf\_input)"; and for ASR, "function parseAsr(asr\_result)". They may each return "\_\_NOMATCH\_\_" to indicate that the interpreter should throw a nomatch event.**

The ASR related properties sensitivity and speedvsaccuracy are supported for Nuance ASR.

If DTMF input has already been received, the interpreter ignores ASR results. This avoids bargein and misrecognition once DTMF input has started in a particular field.



The property 'universals' (VoiceXML 2.0) can be used to add or remove the predefined global grammar elements (help, exit, cancel) from the global grammar.

The property ASRENGINE is used to specify the ASR being used. Currently, the following ASR engines are supported on the given platforms. Multiple ASR engines can be supported at the same time.

```
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
```

The shadow variables `name$.nbestresult` and `name$.nbestconfidence` have been added for `nbest` results when supported by ASR engine. The "nbestresult" gives the ASR `nbest` results in the following format: `res_1|res_2|...|res_n`, where "res\_n" is in the format of `[?][slot1=]val1;[?][slot2=]val2;...`, and where optional things are shown within [...] but '[' and ']' will not be in the string, and a preceding '?' indicates an ambiguous match. The "nbestconfidence" gives the confidence levels for each of the `n` results in the following format: `conf1|conf2|...`

### SpeechWorks OSR1.0 Specific Usage Notes

The W3C XML grammar format is supported. See reference <http://www.w3.org/TR/2001/WD-speech-grammar-20010103/> for more information. Note that the grammar scripting language and semantic interpretation have not been standardized by W3C, thus OSR 1.0 provides its own syntax and interpretation.

1. GARBAGE is treated in the same manner as NULL.
2. Spaces inside <token> are not supported. Leading and trailing spaces will be removed and spaces between words will be treated as “\_”.

For example:

```
<token> New York </token>
```

will be the same as

```
<token> New_York</token>
```



3. `<count number="0+*>`, `<count number="1+*>`, `<count number="?*>` are not supported as yet.
4. The "number" and "currency" builtin speech grammars are not supported in this release.

## URI References

A URI that starts with "builtin:" has special meaning. For audio and script files, this means the file is local to the VoiceGenie platform relative to a platform specific directory. For grammar files, use of the "builtin:grammar/" prefix indicates a grammar file hosted on the VoiceGenie platform. (for ASR engines Nuance, Watson, and MSSR only). Other "builtin:" grammars references imply a grammar that has been preloaded into the ASR engine.

## Session Variables

In addition to standard session variables, namely: `session.telephone.ani`, `session.telephone.dnis`, `session.telephone.iidigits`, and `session.telephone.uui` (note: those are available to first page as well if it is a CGI script/program), we also support the following platform-dependent session variables:

`session.transfer.allow`: assignable only ONCE, and is initially undefined (i.e., transfer allowed); if set to 'NO', no call transfers are allowed for this session. The platform also supports a regular expression rule-based method of allowing or disallowing call transfer based on the DNIS and transfer destination number.

## Default Event Handlers

The default "noinput" event handler does not play prompts, but only reprompts (note this is the correct behavior as per the VoiceXML 1.0 specification.). If you don't like this behavior, you will need to have a noinput handler in your page. The default `<cancel>` has no message and does not reprompt. The default `<help>` event handler on the other hand, should have a prompt (and then reprompt), but it does not have one at this time.



## Logging and Debugging

There is a built-in self-debugging capability for generating email and log information about application execution:

```
<META NAME="MAINTAINER" CONTENT="TECH@VOICEGENIE.COM" />  
<PROPERTY NAME="LOGLEVEL" VALUE="1" />
```

If MAINTAINER is set in the local document, this email address will be used. If not but MAINTAINER is defined in the root document, that email address will be used. Otherwise, no email will be sent. The details of email are controlled by the LOGLEVEL property:

LOGLEVEL	Logged Information
0	no debug information.
1	in addition to level 0, errors (default)
2	in addition to level 1, warnings

There are metrics supported with the server platform to allow users to specify explicitly what depth of information is to be reported: regular page usage (METRICSLEVEL), via

```
<PROPERTY NAME="METRICSLEVEL" VALUE="1" />
```

Note that MASRLEVEL is no longer used and should be considered deprecated. Usage data is kept in a specified metrics file.

The range of METRICSLEVEL is 0 – 3 with the following details:

METRICSLEVEL	Logged Information
0	data on start and end times of calls, and application names if available, as well as the termination reason namely error, user hangup, or application hangup (default)
1	same as level 0.
2	in addition to level 1 reporting, information on prompts and ASR metrics

3

provides information regarding form items visited, etc.

- Grammar load and compile failures are now logged in e-mail messages, and as part of the developer node log. These also cause an `error.application` to be thrown;
- Support for `badfetch.http.nnn` as in VoiceXML 2.0;
- You will receive a warning for variables starting with `'_'`;
- The `<vxml>` tag has a new `LOGPAGE` attribute. When set to `'1'`, it will cause the entire page to be e-mailed to the MAINTAINER.

## Scripting

The VoiceGenie server includes a scripting engine supporting a complete implementation of ECMAScript. Variables in VoiceXML and ECMAScript are equivalent, and are accessible in both realms during execution.

Caveat: when executing in the application root document, the document (`vxml`) level variables can be referenced using the prefix "application", but not "document."

## DTMF Typeahead Behavior

The DTMF input buffer is always cleared on entry to a noinput item such as a `<block>`. Once in a noinput item, all DTMF input will be buffered except for the first `TERMCHAR`, assuming `bargein` is enabled, to support type ahead. On entry to an input item such as a `<field>`, the DTMF input buffer is cleared only if `bargein` is disabled for the first prompt, or if `INPUTMODES` is set to `VOICE` only. When leaving an input item, any unused DTMF in the buffer will be kept. On entry to a new `<form>`, the DTMF buffer will not be cleared unless `CLEARDTMF` attribute is set to "true". Since each prompt in a noinput item can independently be interrupted by DTMF, the input DTMF will be cleared on entry to the next prompt in the same or another noinput item, if any. See notes for `<prompt>` tag below for more info.



If `fetchaudiotime > 0`, `fetchaudiodelay` has no effect. This was done under the assumption that if someone wanted to play audio for a given time, they would not want the delay. The delay was only meant for normal situations where you want to play audio if fetch may take a long time, but you may not want to start the audio and immediately stop it in case the fetch takes only a second or so.

## Speech Markup

Speech markup is supported for the Watson and Speechify TTS engines. Setting the property `TTSENGINE` in the following manner enables it:

```
<property name="TTSENGINE" value="WATSON"/> or;  
<property name="TTSENGINE" value="SSML"/>
```

For Watson, the following markup is supported:

- `<break>`
- `<div>`
- `<pros>` - Ignored
- `<emp>`
- `<sayas>` - sub
- `<sayas>` - PHON is not supported.
- `<sayas class=DIGITS, LITERAL` use Watson's spelling mode, while `PHONE, DATE, NUMBER, CURRENCY, and TIME` use number mode, which uses the text format to generate the audio (for e.g. time should be written as 11:40pm).

The property `TTSENGINE` can be set to either `WATSON` (for the AT&T Watson TTS engine) `SPEECHIFY`, or `SSML` (for the SpeechWorks Speechify TTS engine). The current interpreter supports only a single TTS engine at a time. With the Watson engine, the interpreter will insert Watson specific markup tags in the text prior to delivery to the TTS engine. The tags `<emp>` and `<pros>` may not work with Watson (the proper markup is inserted, but seems to have little effect).

For Speechify, the following markup is supported:



- `<break>`; only the "type" attribute is supported, "msecs" is not;
- `<div>` is supported.
- `<sayas>` is partially supported, attribute "phon" is not supported; attribute "sub" is supported; attribute "class" is supported. However, the date and time classes must follow the formats "yyyy/mm/dd" and "hh/mm" respectively; not following these formats may produce unexpected results.

For the `<sayas>` tag, for class phone, date or time, the TTS text should be in the appropriate format. For phone: nnn-xxx-xxxx; date: `<Month>` dd, yyyy, where `<Month>` is the first 3 characters of the month; time: hh:mmx where x is a, p or h. These formats are currently also required by Speechify. With Speechify or SSML, the interpreter converts the in-line markup to W3C SSML format.

## Audio Controls

Support for audio control using properties VGSKIPHEAD, VGSKIPBACK VGSLOWER, VGFASTER, VGSOFTER, VGLOUDER, VGPAUSE and VGSTOP. VGAUDIOCONTROL property can be used to turn this feature on/off. VGNOAUDIOCONTROL property can be used to fully disable this feature.

These properties can be used to control audio file playback by setting each to a DTMF character. An individual feature can be turned off by setting its property to '-' or null. The property names are suggestive of their behavior. These properties are currently not defined by platform default.

All features can be turned on or off by setting property VGAUDIOCONTROL to "true" (default) or "false". The property VGNOAUDIOCONTROL can be used to disable these features. Once set, it cannot be reset within the same or an enclosed scope.

Note that when audio control mode is active (i.e., the above properties are set properly and an audio is playing), the DTMF input will be used for audio control only. Audio control can be active while bargein is off. When bargein is on, audio control may have unexpected behaviour within a field or menu.



Some final usage notes:

- Audio control only works well in a block, not in field;
- Only one unique DTMF digit (0-9, \*) may be used for each audio control; if the property value is defined as '-' or null, it has no effect.

Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
<assign>	NAME, EXPR				
<audio>	SRC, CACHING, FETCHTIMEOUT, FETCHHINT Supports alternate text and alternate audio			FETCHHINT=STREAM	Support for SRC="...?..." (for query part of URI) New attribute VOLUME (-9 to +9 in dB). EXPR attribute can be used instead of SRC. Contained data can be one of <audio>, PCDATA (for TTS), or <value>, but not more than one. TTS markup is not supported in the content either. TTS markup is not supported within <audio>
<block>	NAME, EXPR, COND				
<break>	MSECS, SIZE				
<catch>	EVENT, COUNT, COND				
<choice>	DTMF, NEXT, EXPR, EVENT, CACHING,			FETCHHINT=PREFETCH	FETCHAUDIODELAY=time (delay for playing



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
	<i>FETCHAUDIO</i> , <i>FETCHHINT</i> , <i>FETCHTIMEOUT</i> Note that <grammar> is not available in <choice>				<i>FETCHAUDIO</i> ; default is 1s) <i>FETCHAUDIOTIME</i> =time (minimum time for playing <i>FETCHAUDIO</i> ; default is 0s) Grammar uses whole choice phrases instead of subset of the choice phrase
<clear>	<i>NAMELIST</i>				
<disconnect>	This element has no attributes.				
<div>	<i>TYPE</i>				
<dtmf>	<i>SRC</i> , <i>SCOPE</i> , <i>TYPE</i> , <i>CACHING</i> , <i>FETCHHINT</i> , <i>FETCHTIMEOUT</i>				Supports full ABNF grammar specification ( <i>MIME</i> type <i>application/x-abnf</i> ), with the exception of optional expansions ( <i>[..]</i> ) and postfix operators ( <i>'+' and '*'</i> ), external grammar imports, special rules <i>\$NULL</i> and <i>\$VOID</i> , and weights Attribute <b>EXPR</b> can be used instead of <b>SRC</b> .
<else>	This element has no attributes.				
<elseif>	<i>COND</i>				
<emp>	<i>LEVEL</i>				



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
<enumerate >	Supported				Content is limited to PCDATA and some tags: <value> with variables '_dtmf' and '_prompt'. No tags, speech markup, or other tags are allowed.
<error>	COUNT, COND				
<exit>	EXPR, NAMELIST				EXPR and NAMELIST do not currently log information.
<field>	NAME, EXPR, COND, TYPE, MODAL, SLOT Supports all shadow variables: name\$.confidence, name\$.utterance (may not be the real utterance depending on ASR engine), and name\$.inputmode Built-in types currency, phone, number, date, digit supported.				Supports extra shadow variables: name\$.bargeinphrase Name\$.bargeinscore And properties: <b>BARGEINLEVEL</b> (=0.0 – 1.0), <b>ASRINTIMEOUT</b> , and <b>ASRENGINE</b> (=NUANCE/WATSON) <b>ENDBEEP</b> – end prompt with a tone (true or false). Built-in types currency, phone, number, digit, time and date require the appropriate grammar files under grammar/<engine> sub directory for Nuance and Watson and MSSR (file name should be <builtin name>.txt). When using builtin grammars with TYPE attribute, the DTMF input or the ASR grammar result



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
					<p>could be processed by a JS under script/builtin directory to make it VoiceXML compliant before assigning the value to the field variable. For DTMF, the file is dtmf_&lt;builtin name&gt;.js; for ASR, the file &lt;engine&gt;_&lt;builtin name&gt;.js, and if it does not exist and a dummy/empty file &lt;engine&gt;_&lt;builtin name&gt;.na also does not exist, default_&lt;builtin name&gt;.js file may be used if it exists.</p> <p>See the notes above for a description of N-best behaviour.</p>
<filled>	<i>MODE, NAMELIST</i>				
<form>	<i>ID, SCOPE</i>				<p>Add new attribute:  <b>CLEARDTMF</b> = TRUE   FALSE (default is FALSE) as based on the VoiceXML specification, DTMF buffer is not cleared on form entry</p>
<goto>	<i>NEXT, NEXTITEM, EXPR, NEXTEXPR, CACHING, FETCHAUDIO,</i>			<i>FETCHHINT=PREFETCH</i>	<p><b>FETCHAUDIODELAY</b>=<i>time</i> (delay for playing <b>FETCHAUDIO</b>; default is 1s)</p>



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
	<i>FETCHHINT</i> , <i>FETCHTIMEOUT</i>				<i>FETCHAUDIOTIME</i> = <i>time</i> (minimum time for playing <i>FETCHAUDIO</i> ; default is 0s) Support <i>NEXT</i> = "... ?..." (for query part of URI)
<grammar>	<i>SRC</i> , <i>SCOPE</i> , <i>CACHING</i> , <i>FETCHHINT</i> , <i>FETCHTIMEOUT</i> <i>TYPE</i>				Support for <i>SRC</i> = "... ?..." (for query part of URI) Type can be: application/x-abnf application/x-voicegenie-nuance application/x-voicegenie-watson Type overrides the content-type returned by Web server. The default type is ABNF (for Nuance). For Watson, only simple in-line grammars are supported. The ASR engine grammar type can be used within CDATA for inline grammars. For global grammars and mixed-initiatives, slot names are required. With ABNF, a slot can be specified within a tag followed by ':' and the tag name (e.g.: from Toronto {from:Toronto} to Halifax {to:Halifax}).
<help>	<i>COUNT</i> , <i>COND</i>				



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
<if>	COND				Nested <i>if</i> supported.
<initial>	NAME, EXPR, COND				ENDBEEP – end prompt with a tone.
<link>	NEXT, EVENT, FETCHAUDIO, FETCHTIMEOUT, FETCHHINT, FETCHAUDIODELAY, FETCHAUDIOTIME, EXPR				
<log>	COND, EXPR, LEVEL				<b>EXPR:</b> An expression which will be evaluated at runtime, and then be appended to the element content in the log. <b>LEVEL:</b> This value has to be less than or equal to the property value <b>LOGLEVEL</b> for the log to be executed.
<menu>	ID, SCOPE, DTMF				<b>DTMF</b> defaults to <i>TRUE</i> .
<meta>	NAME, CONTENT, HTTP-EQUIV				<i>CALLREQUEST=DECLINE</i> rejects the call <i>APPLICATION=appname</i> for billing <i>MAINTAINER=email</i> for email logging.
<noinput>	COUNT, COND				
<nomatch>	COUNT, COND				



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
<object>	<i>NAME, CLASSID, DATA, EXPR, COND, CODEBASE, CODETYPE, TYPE, ARCHIVE, CACHING, FETCHAUDIO, FETCHHINT, FETCHTIMEOUT, FETCHAUDIODELAY, FETCHAUDIOTIME</i>				Note that VoiceGenie currently has no defined object extensions.
<option>	<i>DTMF, VALUE</i>				
<param>	<i>NAME, EXPR, VALUE, VALUETYPE, TYPE</i>				
<prompt>	<i>BARGEIN, COND, COUNT, TIMEOUT</i>	Do not support <i>BARGEIN=TRUE</i> followed by <i>BARGEIN=FALSE</i>			Prompts are now queued until: If one of <exit>, <submit>, <goto>, <transfer>, <subdialog>, <record>, <object> are encountered, then all the prompts will be played, and only DTMF bargein is allowed (i.e., the queue is drained); if prompt <b>BARGEIN</b> is on. Then the <exit>, <submit>, etc. will be executed. If a fetchaudio is specified in the <goto>/<submit>, then the fetchaudio will be played <b>after</b> the queued prompts are



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
					<p>played.</p> <p>If one of &lt;field&gt;, &lt;initial&gt;, &lt;choice&gt;/&lt;menu&gt; is encountered, then all the prompts will be played with the corresponding grammars in the &lt;field&gt;, &lt;initial&gt;, &lt;choice&gt;/&lt;menu&gt; tags enabled. Depending on the prompt <b>BARGEIN</b> attribute/property, the user may be able to <b>BARGEIN</b> while playing prompts.</p>
<property>	NAME, VALUE				<p><b>ASRENGINE</b>: the valid values are Nuance, Watson, Speechworks, MSSR (Microsoft) and LSS, some may not be supported on certain platforms.</p> <p><b>TTSENGINE</b>: the valid values are Watson, Speechify; some may not be supported on certain platforms.</p> <p><b>BARGEINLEVEL</b> confidencelevel required for ASR bargein, range 0 - 1.0; may not be supported by some ASR engines.</p>



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
					<p><b>UNIVERSALS</b> : specify universal commands as in VoiceXML 2.0; value could be one or more of cancel/exit/help, "none" or "all"; default is "all".</p> <p><b>MAXNBEST</b> specify maximum nbest number for ASR (Speechworks and Nuance only)</p> <p><b>VGNOAUDIOCONTROL</b>: once defined to any value, audio control is disabled.</p> <p><b>VGAUDIOCONTROL</b>: "true" enables audio control, "false" disables it.</p> <p><b>VGSKIPHEAD, VGSKIPBACK, VGSLOWER, VGFASTER, VGSOFTER, VGLOUDER, VGPAUSE</b></p> <p><b>VGSTOP</b>: each may specify a DTMF key for the corresponding audio control feature; can be set to '-' or null to disable the particular feature.</p>



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
					<p><b>FETCHAUDIODELAY:</b> delay playing fetchaudio by this time value.</p> <p><b>FETCHAUDIOTIME:</b> play fetchaudio at least this much time.</p> <p><b>LOGLEVEL:</b> controls logging info that could be mailed to "maintainer"; 1 (default) gives only error logging, 2 gives warnings also.</p> <p><b>METRICSLEVEL:</b> controls info logged to metrics file which can be used for application tracing or debugging; 0 or 1 (default) gives only call start and end traces, 2 gives prompts and ASR results also, and 3 gives more info such as each form item visited.</p>
<pros>	<i>VOL</i>				<b>PITCH, RATE, RANGE</b> ignored
<record>	<i>NAME, EXPR, COND, MODAL=TRUE, BEEP, MAXTIME,</i>			<grammar> element inside <record>	<b>BEGINSILENCE</b> (duration of <i>initial</i> silence before terminating recording –



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
	<p><i>FINALSILENCE</i>, <i>DTMFTERM</i>, <i>TYPE</i> Supports all shadow variables: <i>name\$.duration</i>, <i>name\$.size</i> (both may not be accurate), and <i>name\$.termchar</i> <i>TYPE</i> supports audio/basic (default and same as audio/vox) and audio/wav, and also audio/adpcm or audio/adpcm8, and audio/adpcm6</p>				<p>default is 4s, <b>minimum is 100ms, maximum is 600s</b>); <i>MINTIME</i> (if recording duration is less than this, <b>assume empty recording, throw noinput</b>; default/<b>minimum is 250ms, maximum is 600s</b>); <i>AGC</i> (automatic gain control; default is <i>TRUE</i>); <i>FINALSILENCE</i> (default is 4s, <b>minimum is 250ms, maximum is 100ms</b>) Added attribute <i>AGC</i> (automatic gain control), default is on. Attribute <b>TYPE</b> supports audio/basic (default and same as audio/vox) and audio/wav, and also audio/adpcm or audio/adpcm8, and audio/adpcm6. Shadow variables <i>name\$.duration</i> and <i>name\$.size</i> may not be accurate.</p>
<reprompt>	This element has no attributes. Reprompt is only used to clear the prompt flag for the next				



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
	form item that will be visited. This is relevant only if the last form item had resulted in a catch event				
<return>	<i>EVENT, NAMELIST</i>				
<sayas>	<i>CLASS</i>				<b>PHON, SUB</b> ignored
<script>	Fully ECMA script compliant. <i>SRC, FETCHTIMEOUT</i>				Variables are accessible across the boundary of script and markup parts. Scopes are handled properly and no extra copying is needed
<subdialog >	<i>NAME, EXPR, EXPR, COND, NAMELIST, METHOD, CACHING, ENCTYPE, FETCHAUDIO, FETCHTIMEOUT, FETCHHINT, FETCHAUDIODELAY, FETCHAUDIOTIMEOUT</i>	<i>MODAL=FALSE</i>			Attribute <b>SRCEXPR</b> has been added to specify the URI as an ECMAScript expression instead of using attribute <b>SRC</b> .
<submit>	<i>NEXT, EXPR, NAMELIST, METHOD, ENCTYPE, CACHING, FETCHAUDIO, FETCHHINT, FETCHTIMEOUT</i>			<i>FETCHHINT=PREFETCH</i>	<i>FETCHAUDIODELAY=time</i> (delay for playing <i>FETCHAUDIO</i> ; default is 1s) <i>FETCHAUDIOTIME=time</i> (minimum time for playing ) Support <i>NEXT="...?..."</i> (for query part of URI) if <i>METHOD="GET"</i>



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
<throw>	<i>EVENT</i>				
<transfer>	<i>NAME, EXPR, COND, DEST, DESTEXPR, BRIDGE, CONNECTTIMEOUT, MAXTIME</i> Supports all shadow variable: <i>name\$.duration</i> and event: <i>telephone.disconnect.hangup</i> Prompt counter is also supported (primarily for <goto>)			<grammar> inside <transfer>  Attribute <b>ANALYSIS</b> ("true" or "false" (default)), and <b>CONNECTWHEN</b> ("analysis" or "answered" or "immediate"(default)). Note that <b>CONNECTWHEN</b> must be 'answered' when using analog or robbed-bit T1 configurations. The <dtmf> tag can be used within a <transfer> to specify DTMF that will terminate a transfer. <i>The minimum for MAXTIME is 30s, maximum is one week, and the default, same as MAXTIME=0, is no limit. Attribute UUIDATA is used to pass value to theoutbound call (default is to send the UUIDATA from the inbound line).</i>	DEST is formatted as "phone://(416)736-0905x111" where brackets and '-' are optional.
<value>	<i>EXPR, MODE, CLASS, MODE</i>				For playing back recordings, <i>MODE="recorded"</i> is required
<var>	<i>NAME, EXPR</i>				
<vxml>	<i>VERSION, BASE,</i>			Note that "LANG" is	Use <i>VARname</i> instead of



Element	Required Attributes		Optional Attributes		Extensions
	Supported	Unsupported	Supported	Unsupported	
	<i>APPLICATION, LANG</i>			ignored.	<i>APPLICATION.VARname</i>  <i>New Attribute LOGPAGE – this will e-mail the current page (and root page, if any) to the MAINTAINER, if set to '1'</i> <i>APPLICATION can refer to a URI with parameters.</i>