

VOICEGENIE

VoiceGenie 7 Application Migration Guide

April 13th, 2005



VoiceGenie Technologies Inc.
1120 Finch Ave. W. • Toronto, Ontario • M3J 3H7 • Canada
T. +1.416.736.4151 • F. +1.416.736.1551 • support@voicegenie.com
www.voicegenie.com

VoiceGenie Contacts

VoiceGenie Technologies Inc.
1120 Finch Avenue West
Toronto, Ontario
Canada
M3J 3H7

T. +1.416.736.4151
F. +1.416.736.1551
support@voicegenie.com

<http://www.voicegenie.com/index.html>

Proprietary / Copyright Information

This material contains proprietary and/or copyright information of VoiceGenie Technologies Inc. and may not be copied, used, or disclosed without the permission of VoiceGenie Technologies Inc.

© COPYRIGHT 2005 VoiceGenie Technologies Inc.

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, this information is subject to change without notice.

Please note that VoiceGenie makes no warranties with respect to future releases. The information provided herein is for informational purposes only. VoiceGenie reserves the right to change product release schedules and/or features allocated to a product release at any time.

Trademarks

All trademarks are the property of their respective owners. Where those designations appear in this document, and VoiceGenie is aware of a trademark claim, the designations have been printed in initial caps or all caps.



Table of Contents

Proprietary / Copyright Information	3
Notice	3
Trademarks	3
Table of Contents	4
Revision History	6
1 Introduction	7
2 VoiceXML 2.1 Features	7
2.1 <data> Element	7
2.2 <mark> Element	7
2.3 <foreach> Element	7
2.4 RECORDUTTERANCE Property	8
2.5 <i>srcexpr</i> Attribute in <grammar> and <script> Elements	8
3 VoiceXML Conformance	8
3.1 Call analysis is TRUE by default	8
3.2 URLs are resolved relative to the document in which an event was thrown	9
3.3 error.unsupported.builtin event	9
3.4 application.lastresult\$.length <= MAXNBEST	9
3.5 session.connection.protocol.<protocol name> must be defined	9
3.6 Actual timeout elapsed for prompts cannot be less than specified prompt timeout	9
3.7 Timeouts for empty prompts won't be ignored	9
3.8 Mixed barge-in support	10
3.9 Dialog event counters are reset when the dialog is transitioning to itself	10
4 SSML 1.0 Support	10
5 Logging Changes	10
5.1 Enhancements to Existing Metrics Entries	10
5.1.1 <i>goto</i>	10
5.1.2 <i>subdialog_start</i>	11
5.1.3 <i>form_select</i>	11
5.1.4 <i>record_end</i>	12
5.1.5 <i>wf_arrived, fetch_end</i>	13
5.1.6 <i>input_start</i>	13
5.2 New Metrics Entries	14
5.2.1 <i>submit</i>	14
5.2.2 <i>root_appl</i>	14

5.2.3	<i>filled_exit</i>	14
5.2.4	<i>choice_select</i>	15
5.2.5	<i>event_handler_enter</i>	15
5.2.6	<i>event_handler_exit</i>	16
5.2.7	<i>filling</i>	16
5.2.8	<i>dtmf_end</i>	16
6	Other Changes	17
6.1	New Instance ID Format	17
6.2	Including Session ID in Maintainer E-mails	17
6.3	connectwhen default value	17
6.4	Undefined shadow variables	17
6.5	The application.lastresult\$ shadow variable shouldn't be read-only	17
6.6	Empty src/expr/srcexpr attributes in <script> and <grammar>	18
6.7	Undeclared variable name in <submit> namelist	18
7	New Features	18
7.1	Conferencing Support with <join>	18
7.2	TDD/TTY Support	18
7.3	SIP Media Redirection Support	18
7.4	_VGGetInfo() function	18

Revision History

Version	Date	Change Summary	Author/Editor
0.1	January 14 th , 2005	Initial Draft	Ates Goral
0.2	January 17 th , 2005	Minor additions, incorporated Katie Wang's suggested changes	Katie Wang
0.3	February 10 th , 2005	Proofing	Rob Marchand
0.4	February 18 th , 2005	Proofing, removed session.connection.aai section, incorporated Laura Sasaki's suggested changes	Ates Goral
0.5	March 1 st , 2005	Additions to the "Other Changes" section	Ates Goral
1.0	April 13 th , 2005	Revised for VoiceGenie 7 Release	Andrew Ho

1 Introduction

This document outlines the new features and changes in the VoiceGenie 7 release that may require changes to existing VoiceXML applications.

2 VoiceXML 2.1 Features

The following VoiceXML 2.1 features are only made available if the `<vxml>` *version* attribute is specified as 2.1 (or higher¹):

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

Support for the *namelist* attribute in `<disconnect>` ([VoiceXML 2.1 Specification, section 8](#)) and the *type* attribute in `<transfer>` ([VoiceXML 2.1 Specification, section 9](#)) existed prior to VoiceGenie 7, as VoiceGenie extensions.

See the [VoiceXML 2.1 specification](#) for more information.

2.1 <data> Element

See the [VoiceXML 2.1 Specification, section 5](#) for a description of this feature.

When the `<vxml>` *version* attribute is specified as less than 2.1, the `<data>` element is not recognized. It will be treated as an invalid child element.

2.2 <mark> Element

See the [VoiceXML 2.1 Specification, section 4](#) for a description of this feature.

When the `<vxml>` *version* attribute is specified as less than 2.1, the `<mark>` element is not recognized. It will be treated as an invalid child element.

2.3 <foreach> Element

See the [VoiceXML 2.1 Specification, section 6](#) for a description of this feature.

VoiceGenie has previously supported the `<foreach>` element as a proprietary extension, however only in executable content. VoiceXML 2.1 defines the `<foreach>` element within executable content as well as a child of `<prompt>`. When the `<vxml>` *version* attribute is specified as 2.1 (or higher), the `<foreach>` element will be allowed as a child of `<prompt>`. Otherwise it will be treated as an invalid child element.

¹ Currently, the latest VoiceXML specification is at version 2.1. Specifying the `<vxml>` *version* attribute as higher than 2.1 (e.g. 2.5, 3.0) will be of the same effect as specifying it as 2.1.

Note: When <foreach> is inside a <prompt>, elements allowed in executable content are not allowed within <foreach>.

2.4 RECORDUTTERANCE Property

See the [VoiceXML 2.1 Specification, section 7](#) for a description of this feature.

When the <vxml> *version* attribute is specified as less than 2.1, the RECORDUTTERANCE property will be ignored.

VoiceGenie previously introduced a proprietary SAVEUTTERANCE attribute having the same functionality. This will continue to be supported. If the <vxml> *version* attribute is specified as 2.1 or greater, if either the RECORDUTTERANCE or the SAVEUTTERANCE attribute is set to true, then the user's utterance for the input item is recorded. Furthermore, there is a VoiceGenie extension property UTTERANCEDEST which is used to specify the disk location of the recorded file. This will be supported for the RECORDUTTERANCE property as well.

2.5 *srcexpr* Attribute in <grammar> and <script> Elements

See the [VoiceXML 2.1 Specification, sections 2 and 3](#) for a description of these features.

When the <vxml> *version* attribute is specified as less than 2.1, the *srcexpr* attribute will be ignored. VoiceGenie has existing support for the *expr* attribute for referencing <grammar> and <script> files dynamically. The *expr* attribute should be used for these documents.

When the <vxml> *version* attribute is specified as 2.1 (or higher), the *expr* attribute is not allowed for <script> and <grammar>. If the *expr* attribute is used, an error.badfetch event will be thrown. For these documents, the new *srcexpr* property should be used.

3 VoiceXML Conformance

In addition to the availability of the new VoiceXML 2.1 features, the COM.VOICEGENIE.STRICTCONFORMANCE property is always TRUE when the <vxml> *version* attribute is specified as 2.1 (or higher). Setting it to FALSE through a <property> element has no effect.

The following sections describe the behavioural changes in VoiceGenie 7 that take effect when the COM.VOICEGENIE.STRICTCONFORMANCE property is set to TRUE or the <vxml> *version* attribute is specified as 2.1 (or higher).

See the section on "Conformance" in the *VoiceGenie 7 Platform Reference* for other behavioural changes when this property is set to TRUE.

3.1 Call analysis is TRUE by default

The *analysis* attribute for <transfer> will be TRUE by default.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, the default for *analysis* is FALSE.

3.2 URLs are resolved relative to the document in which an event was thrown

If an event is caught in a different page (application root), relative URLs are resolved relative to the URL of the page that the event was thrown in.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, relative URLs are resolved relative to the page that catches the event.

3.3 error.unsupported.builtin event

For an invalid *type* attribute value for <field>, the error.unsupported.builtin event is thrown at runtime.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, the event is thrown at compile time and therefore cannot be caught by the page.

3.4 application.lastresult\$.length <= MAXNBEST

The length of the application.lastresult\$ array will not be longer than what is specified in the MAXNBEST property even when more than MAXNBEST results are available from recognition.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, application.lastresult\$ can have more than MAXNBEST results.

3.5 session.connection.protocol.<protocol name> must be defined

The *session.connection.protocol.<protocol name>* session variable specifies the name of the underlying connection protocol. A corresponding session variable property referring to a JavaScript object will be defined. For example, if the protocol name is H323, *session.connection.protocol.h323* is defined as a JavaScript object. This object may or may not have additional properties depending on the protocol being used.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, this session variable is not always defined (depending on the protocol being used).

3.6 Actual timeout elapsed for prompts cannot be less than specified prompt timeout

When the actual time elapsed during a timeout is measured (e.g. By using JavaScript in the page) it will never be less than the specified timeout. It may be slightly longer due to the design of the platform.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, the actual timeout elapsed may be slightly shorter than the intended amount.

3.7 Timeouts for empty prompts won't be ignored

If the <prompt> tag is empty (i.e. doesn't contain any children elements or text), its timeout will still be processed.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, the timeouts of empty or invalid prompts are not taken into account.

3.8 Mixed barge-in support

Prompts with different barge-in values can be used without limitations in input items.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, within a certain input item, once a prompt with barge-in set to TRUE is used, the prompts following that prompt will all be assumed to have barge-in set to TRUE.

3.9 Dialog event counters are reset when the dialog is transitioning to itself

When a dialog is transitioning to itself, its event counters will be reset.

With COM.VOICEGENIE.STRICTCONFORMANCE set to FALSE, the event counters will be preserved.

4 SSML 1.0 Support

The tags <lexicon>, <metadata> and <meta> as defined by SSML 1.0 are now supported. Prior to VoiceGenie 7, they would be treated as invalid child elements.

See the [SSML 1.0](#) specification for more information.

5 Logging Changes

The format of some of the existing metrics log entries has been enhanced, and some new metrics entries have been added. These changes should be taken into account for external applications that perform metrics file processing.

5.1 Enhancements to Existing Metrics Entries

5.1.1 goto

The *goto* metric was previously logged only when transitioning to a dialog in the *current* page. When transitioning to a dialog on a different page, the URL of the page and the target dialog will now also be logged. This entry is logged for <goto>, <link> or <choice>.

The new format for the *goto* metric entry is:

```
goto :[<target URL>][#<dialog ID>[.<form item name>]]
```

Example 1: Transitioning to a different page

```
goto :http://host.com/page.vxml
```

Example 2: Transitioning to a dialog on a different page

goto :http://host.com/page.vxml#address_form

Example 3: Transitioning to a dialog on the current page

goto :#address_form

Example 4: Transitioning to a form item on the current page

goto :#address_form.city

5.1.2 subdialog_start

The subdialog_start metric previously included only the name of the subdialog and the submitted parameters. Now, the URL of the subdialog will also be logged.

The new format for the subdialog_start metric entry is:

subdialog_start :<subdialog name>:<target> |param|[name1;value1;[name2;value2];...]

Where **target** has the following format:

[<target URL>][#<dialog ID>[.<form item name>]]

Examples:

subdialog_start :multiply:http://host.com/calc.vxml#multiply |param|a;5;b;3;

subdialog_start :login:#login |param|user;bob;pass;secret;

subdialog_start :login:http://host.com/next.vxml |param|field1;42;

5.1.3 form_select

The form_select metric now includes a parameter that explicitly specifies the type of form item that is selected by the FIA.

The new format for the form_select metric entry is:

form_select :<form item name>:<form item type>

Where **form item type** can be one of the following:

FIELD
BLOCK
TRANSFER
RECORD
SUBDIALOG
INITIAL
OBJECT

Example:

form_select :city:FIELD

5.1.4 record_end

The reason for the termination of a <record> will also be logged.

The new format for the form_select metric entry is:

record_end :<outcome>[<success reason>|<duration>|<audio format>|<recording filename>]

The **outcome** can be one of the following:

RECORD SUCCESS
NOINPUT
local grammar match
global grammar match

NOINPUT indicates that there was no audio input within the timeout period.

When recording is terminated by DTMF input and a local or global grammar match (*dtmfterm* is set to FALSE) before there was audio input, a *local grammar match* or *global grammar match* is logged.

RECORD SUCCESS indicates that there was audio input before record termination and the <record> element is being filled. In this case, **duration**, **audio format** and **recording filename** are also logged and the **success reason** can take on the following values.

DTMF
MAXTIME
FINALSILENCE

DTMF indicates that the recording was terminated by DTMF input while *dtmfterm* was set to TRUE.

MAXTIME is logged when the recording was terminated because the audio input exceeded *maxtime*.

FINALSILENCE means that the audio input was terminated by a period of silence as specified in *finalsilence*.

Examples:

record_end :RECORD SUCCESS|DTMF|4|audio/vox|usr/local/tmp/file0001.vox

record_end :RECORD SUCCESS|MAXTIME|20|audio/wave|usr/local/tmp/file0002.vox

record_end :NOINPUT

record_end :global grammar match

5.1.5 wf_arrived, fetch_end

These metrics entries have been extended to provide information on how the fetch was performed (for successful fetches only). **wf_arrived** is logged for VoiceXML page fetches and **fetch_end** is logged for all other types of fetches (grammar, audio etc.).

The new formats of these entries are:

```
wf_arrived s (<origin>):<full URI>
fetch_end done (<origin>):<full URI>
```

Where **origin** can be one of the following:

```
memory
proxy-hit
proxy-miss
direct
file
```

memory means that the file was served from the shared memory cache of the Fetching Module.

proxy-hit means that the file was served from the HTTP proxy (Squid).

proxy-miss means that the HTTP proxy didn't have a fresh enough copy and the file was fetched from the application server by the HTTP proxy.

direct indicates a direct fetch from an application server when the platform is configured to bypass the HTTP proxy.

file is logged for local files (file://...).

Examples:

```
wf_arrived s (proxy-hit):http://host.com/page.vxml
wf_arrived s (file):file:///usr/local/pages/test.vxml
fetch_end done (memory):http://host.com/audio/beep.vox
```

5.1.6 input_start

This metrics entry is extended to include the active input modes during recognition.

The new format for this entry is:

```
input_start :[VOICE][[DTMF][[TDD]
```

The different input modes are in no particular order. The input modes are delimited by pipe ('|') characters.

Example:

input_start :VOICE|DTMF

5.2 New Metrics Entries

5.2.1 submit

When a <submit> is executed, the URL being transitioned is logged along with submitted parameters, if any.

The format of this new entry is:

```
submit :<target> |namelist[[name1;value1;[name2;value2];...]
```

Where **target** has the following format:

```
[<target URL>][#<dialog ID>[.<form item name>]]
```

Examples:

```
submit :http://host.com/page.cgi#entry |namelist|app;45;page;3;
```

```
submit :#dialog2 |namelist|
```

5.2.2 root_appl

Whenever a page is first entered, if the *application* attribute of the <vxml> element is set, the full URI of the root document is logged.

The format of this new entry is:

```
root_appl :<full URI>
```

Example:

```
root_appl :http://host.com/apps/root.vxml
```

5.2.3 filled_exit

This entry is logged when the execution of a <filled> element is complete. This means that the closing </filled> tag was reached and the <filled> was not left because of a <goto>, <submit>, <throw>, <return>, <exit>, <disconnect> or an event.

The format of this new entry is:

```
filled_exit
```

5.2.4 choice_select

When a <choice> in a <menu> is selected, the choice phrase and the choice action are logged. The choice action can be a target URL (and possibly an anchor specifying a dialog on that page) of a page that is being transitioned to or an event that is being thrown.

The format of this new entry when <choice> uses *next* or *expr* is:

```
choice_select :[<DTMF digits>][[<PCDATA of selected choice>] next=<target>
```

Where **target** has the following format:

```
[<target URL>][#<dialog ID>[.<form item name>]]
```

The format of this new entry when <choice> uses *event* or *eventexpr* is:

```
choice_select :[<DTMF digits>][[<PCDATA of selected choice>] event=<event name>
```

Example 1: <choice> with next

```
choice_select :3|three next=http://host.com/page.vxml#address_form
```

Example 2: <choice> with event

```
choice_select :1|one event=one_selected_event
```

5.2.5 event_handler_enter

This entry will be logged whenever an event handler is entered due to an event being caught.

The format of this new entry is:

```
event_handler_enter :<event name>|<event handler location>
```

Where **event handler location** points to the location in a document where the event handler resides. It has the following format:

```
[<target URL>][#<dialog ID>[.<form item name>]]
```

The **event name** can be the *event* attribute of the <catch> element or the tag name of a <nomatch>, <noinput>, <error> or <help> element. This may or may not be the same as the name of the actual event that is caught:

```
error.connection  
NOINPUT  
HELP
```

Examples:

```
event_handler_enter :NOINPUT|http://host.com/root.vxml
```

```
event_handler_enter :myevent|http://host.com/page.vxml#dialog2.field3
```

5.2.6 event_handler_exit

Whenever the execution of an event handler is complete, this entry is logged. This means that the closing `</catch>`, `</noinput>`, etc. tag was reached and the event handler was not left because of a `<goto>`, `<submit>`, `<throw>`, `<return>`, `<exit>`, `<disconnect>` or an event.

The format of this new entry is:

```
event_handler_exit :<event name>
```

The **event name** follows the same rules as in 5.2.5.

Example:

```
event_handler_exit :error.connection
```

5.2.7 filling

Whenever an input item gets filled with a value, the name of the form item along with its new value is logged.

The format of this new entry is:

```
filling :<form id>.<input item name>:<input item type>:<value>
```

Where **input item type** can be one of the following:

```
FIELD  
TRANSFER  
RECORD  
SUBDIALOG  
OBJECT
```

5.2.8 dtmf_end

This is logged when the collection of DTMF digits ends following a **dtmf_start** entry in DTMF mode.

The format is:

```
dtmf_end :<reason>
```

Where **reason** can take on the following values:

```
MATCHED  
NO_MATCH
```


NO_INPUT
NO_DTMF_GRAMMAR
GRAMMAR_ERROR
INTERNAL_ERROR
ABORTED

Example:

```
dtmf_end :MATCHED
```

6 Other Changes

6.1 New Instance ID Format

The format of the *session.com.voicegenie.instance.myself* session variable has been changed to the same as the active session ID. It will no longer contain the platform IP address. The feature described in Section 7.4 can be used to get the platform IP address.

Instance ID example:

```
000B0006-1004A6BE-0001
```

6.2 Including Session ID in Maintainer E-mails

The first line of a maintainer e-mail body will be as follows:

```
Session_ID: <session ID>
```

Where session ID is the active session ID. For example:

```
000B0006-1004A6BE-0001
```

6.3 connectwhen default value

The <transfer> *connectwhen* attribute will default to “answered” instead of “immediate” when *transferaudio* is specified.

6.4 Undefined shadow variables

Previously, some shadow variables were being set to the JavaScript string “undefined”. When a shadow variable is not set, its value will be the JavaScript value *undefined* instead.

6.5 The application.lastresult\$ shadow variable shouldn't be read-only

The *application.lastresult\$* shadow variable can now be modified by a VoiceXML application. Prior to VoiceGenie 7, this variable was read-only and changing the values of its members would have no effect.

6.6 Empty *src*/*expr*/*srcexpr* attributes in `<script>` and `<grammar>`

When the *src*, *expr* or *srcexpr* attributes were declared as empty strings, the `<script>` and `<grammar>` elements were treated as inline scripts and grammars respectively, even though they had no content. Now, the *src*, *expr* or *srcexpr* attributes will be taken into account even when they are declared as empty strings and an external script/grammar fetch will be attempted (the resulting absolute URL will likely be an invalid URL and this will result in an *error.badfetch* event being thrown).

6.7 Undeclared variable name in `<submit>` *namelist*

When an undeclared variable name is used in the *namelist* attribute in `<submit>`, an *error* event was being thrown. With VoiceGenie 7, an *error.semantic* event will be thrown.

7 New Features

7.1 Conferencing Support with `<join>`

The *from*, *to*, *talk* and *listen* attributes of the `<join>` element can now point to conference identifiers, which are unique identifiers prefixed with "conf:". This does not affect the existing `<join>` element functionality.

7.2 TDD/TTY Support

The INPUTMODES property can now also take the value "TDD" as one of the tokens.

The `<grammar>` *mode* attribute can be "TDD". Consequently, the \$.inputmode shadow variable can also be "TDD" following recognition.

Two new properties have been introduced: COM.VOICEGENIE.TDDINTERDIGITTIMEOUT and COM.VOICEGENIE.TDDTERMTIMEOUT.

This new feature does not affect other input modes or grammar modes.

7.3 SIP Media Redirection Support

The type attribute of the `<join>` element can now also be "mediaredirect" indicating that this is a SIP media *redirection join*. This type of transfer will release the media stream from the platform, while maintaining the call control (SIP messaging) under platform control. An attempt to use media redirection along with the TLT feature will result in an application error.

7.4 `_VGGetInfo()` function

A new global JavaScript function, `_VGGetInfo()` is introduced to allow applications to access information about the platform and the running page, as well as custom parameters that can be defined in the VoiceXML Interpreter configuration. The `_VGGetInfo()` function takes a string argument and returns a string value or an empty string if the specified parameter name is not recognized.

`_VGGetInfo('host_ip')` returns the platform IP address
`_VGGetInfo('host_name')` returns the platform hostname

_VGGetInfo('running_uri') returns the URI of the current VoiceXML page