



## Version 1.0

# Frequently Asked Questions

## General

### ***What is Voiyager?***

Voiyager is a productivity platform for VoiceXML applications with Version 1.0 of Voiyager focusing on the complete development and testing of VoiceXML applications. Fully exercising a VoiceXML application, Voiyager supplies the components to support the following VoiceXML evaluations.

- Systematically identifies application defects
- Methodically discovers common usability problems
- Thoroughly validates call flow and business logic

Voiyager also provides an interactive *Browser Phone* to allow manual testing of VoiceXML applications without the need of a telephony platform.

### ***How does Voiyager work?***

Voiyager successfully takes the place of a voice browser with the additional capacity to determine what specific audio clips play, the active grammars, and the valid choices that a caller will speak or enter. With this information, Voiyager identifies dialog states and navigates the complete application. In addition, Voiyager places numerous virtual calls into the VoiceXML application and tests all possible flow paths demonstrating that Voiyager understands the application call flows. As Voiyager evaluates the call flows, Voiyager detects and identifies any common application faults.

### Question Areas Included

General	1
Support and Compatibility	1
Features	3
Purchasing and Licensing	7

## Support and Compatibility

### ***What VoiceXML version(s) does Voiyager support?***

Voiyager supports VoiceXML versions 2.0 and 2.1.

### ***What grammar formats does Voiyager support?***

Voiyager supports SRGS (W3C Recommendation 16 March 2004) XML.

Voiyager also supports the following forms of Semantic Interpretation within SRGS grammars.

- SISR 1.0(W3C Recommendation 5 April 2007)
- Nuance GSL Semantic Interpretation

*Voiyager does not currently support SRGS ABNF syntax.*

### **What built-in grammars does Voiyager support?**

Voiyager uses English language built-ins as defined in section 2.3.1.1 of the VoiceXML 2.0 specification. The Voiyager list includes the following built-ins.

- boolean
- date
- digits
- currency
- number
- phone
- time

### **What audio formats does Voiyager support?**

Voiyager supports the following audio formats.

- 
- |                                      |   |
|--------------------------------------|---|
| • Mime-type:audio/basic              | muLaw, 8000 samples per second  |
| • Mime-type:audio/x-alaw-basic       | ALaw, 8000 samples per second   |
| • Mime-type:audio/x-syntellect-basic | Dialogic, 4 bit ADPCM, 8000   |
| • Mime-type:audio/wav or audio/x-wav | Windows compatible .wav files. Audio passes to the Windows client in the form provided by the application. Note also that if a dialog contains multiple audio files with different audio sample rates, all audio plays using the first audio file rate. |
- 

### **Does Voiyager support Speech Synthesis Markup Language (SSML)?**

Voiyager does not directly process Speech Synthesis Markup Language (SSML) and, therefore, is not impacted by the presence or absence of SSML.

### **How does Voiyager handle platform specific extensions?**

Voiyager implements customizations for compatibility with popular VoiceXML platforms. Voiyager ignores extra VoiceXML tags and attributes as well as unknown Object tags.

*Please contact the Voiyager sales team to confirm support for your platform feature needs.*

### **How does Voiyager manage dynamically generated VoiceXML?**

Voiyager contains a VoiceXML browser to process both static and dynamically generated VoiceXML in the same manner when the VoiceXML content is hosted by a Web or application server.

### **How does Voiyager exercise sub-dialogs?**

Voiyager properly follows sub-dialog references and processes sub-dialogs according to the VoiceXML 2.x specification.

***Does Voyager support shadow variables?***

The Voyager project definition editor allows users to specify application specific shadow variables that are set at the beginning of each call.

***How does Voyager handle dynamically generated grammars?***

Voyager supports both static and dynamically generated grammar content.

***Does Voyager cache data?***

Within a single exploration or application comparison, Voyager will cache audio and grammar resources. Voyager will not cache individual Browser Phone calls or across different explorations or comparisons.

***What languages does Voyager support?***

Voyager explores applications independent of the application's language, however, the Voyager user interface is currently only available in English.

*Please contact Voyager Product Marketing to explore translation to meet your language needs.*

***How does Voyager process host data?***

A common misconception is that Voyager only simulates how your VoiceXML application runs. Voyager does not *simulate* your VoiceXML application—Voyager actually executes your complete application.

Voyager incorporates a VoiceXML browser that implements your full application. All host data and your complete business logic are accessed and executed using the same approach as calls placed into your telephony platform.

## Features

***What types of problems does Voyager Exploration detect?***

Voyager's dynamic and powerful functionality detects problems in the following areas.

**VoiceXML Errors**

- VoiceXML Syntax Errors
- SRGS Syntax Errors
- Missing VoiceXML and SRGS Documents
- Missing Audio Resources

**Possible Problems**

- Application Hang-ups
- Application Transfers
- Infinite Call Loops

**User Defined Problems**

- Audio Clip Played
- TTS Text Played
- Transfer Target Reached

### ***What types of changes does Voyager Comparison detect?***

Voyager's dynamic and powerful functionality also detects changes in the following areas.

#### **VoiceXML Errors**

- VoiceXML Syntax Errors
- SRGS Syntax Errors
- Missing VoiceXML and SRGS Documents
- Missing Audio Resources

#### **Application Changes**

- Changes in Call Paths
- Changes in Audio Files
- Changes in Grammar Files

### ***How do I integrate Voyager into my build and test environment?***

Voyager includes both .net and Java APIs allowing developers to perform all features available to the user through the Voyager client. This includes, but is not limited to the following functionality.

- Starting and monitoring explorations and comparisons
- Evaluating exploration and comparison results
- Generating reports
- Directing Browser Phone calls to test specific use cases

### ***What are the run times for Voyager Explorations and Comparisons?***

Voyager exploration and run times are application specific but the following use case provides insight into common time expectations.

Without application host delays or significant delays fetching the application resources, Voyager typically places 50 calls per minute with 25 concurrent calls occurring.

An average 40 state application is explored with 500 calls. If the application requires test accounts, each account is likely to require 500 calls. Therefore, performing an exploration on this application, with 10 test accounts, is expected to take approximately 100 minutes or nearly two hours. Most applications, however, experiences some amount of host delay validating the accounts and retrieving other data so this value is considered a best case expectation.

### ***How does Voyager determine valid account numbers and pins?***

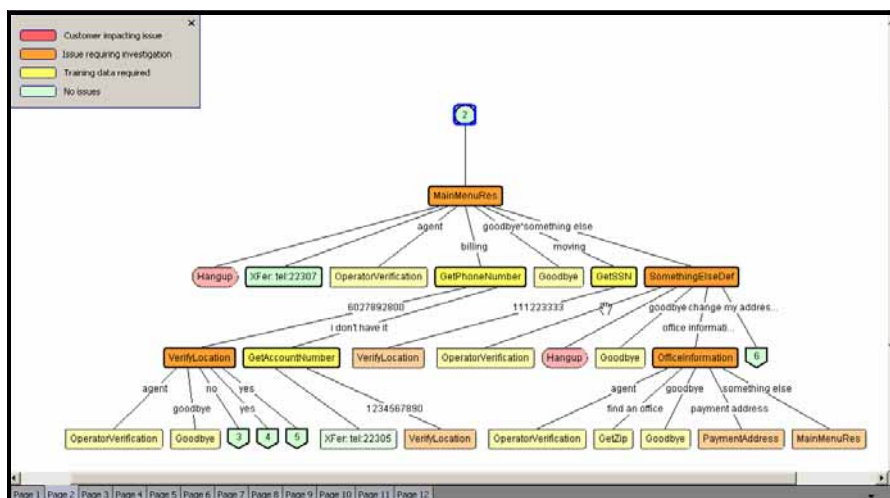
When Voyager encounters a grammar with a large number of unique inputs, such as account or pin input, Voyager randomly chooses valid grammar values and raises an issue indicating that training data is needed. The tester then provides data through the Training Collection editor.

	GetPhoneNumber	GetAccountNumber	GetZip	GetSSN		
<b>Record 1</b>	6027892800	1234567890	85340	111223333		
<b>Record 2</b>	i don't have it	1234567890	85340	111223333		
<b>Record 3</b>	6027892800	I don't have it	85340	111223333		

### How does Voyager know a call flow is correct ?

Voyager cannot automatically determine if a call flow or the business logic in an application is correct, however, Voyager provides two components that allow a developer or analyst to validate an application quickly and easily.

- A call flow diagram displays a visual view of the application.



- A **State Validator** allows a user to review the call flow through each individual dialog state. The user quickly sees how a state responds with different training data and different caller responses.

States:

Training:

Record Name	GetPhoneNumber	GetAccountNumber	GetZip	GetSSN
Record 1	6027892800	1234567890	85340	111223333
Record 2	I don't have it	1234567890	85340	111223333
Record 3	6027892800	I don't have it	85340	111223333

Prompt Scenario:

Phrases:

Calls:

States: 40    Training Records: 3    Prompt Scenarios: 5    Phrases: 4    Calls: 2

---

**Application:** ☎  
 All right I can help with your bill. First can I get your home telephone number area code first. If you do not have it just say I do not have it

**Client:**  
 6027892800

**Application:** ☎  
 Thank you. Let me just pull up your account Here it is. Now, just to make sure that I have the right one, I show that the street number of the account is one two three four and the ZIP code is eight five zero five three Do I have that right

### How is the Voyager Browser Phone different than a VoIP phone?

VoIP and other telephony-based calls require a developer or analyst to interact with an application at the same speed as a caller. Voyager interfaces directly the application bypassing the telephony and the Automatic Speech Recognition (ASR) layer. This gives the Voyager Browser Phone the advantage to offer several features not available with VoIP.

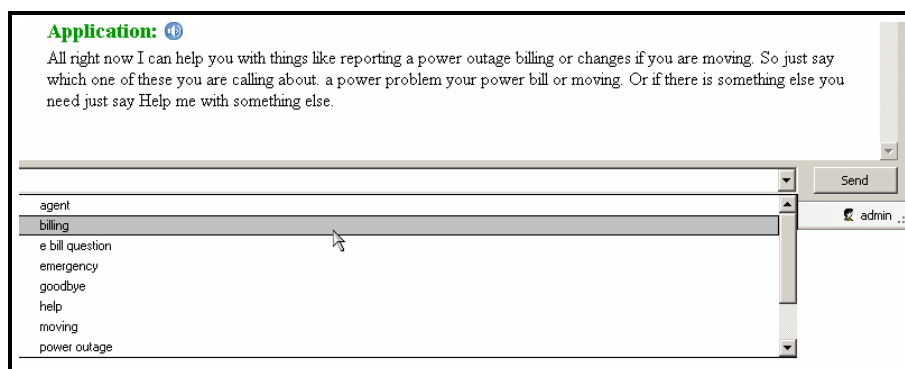
## Input Accuracy

When testing Speech Recognition applications, voice recognition servers periodically fail to recognize a correct interaction. This failure leads to test confusion as the tester performing the test attempts to determine if the failure was due to a speech recognition error or a logic error within the application.

The opposite is also a problem, the tester providing invalid input or no input and yet the recognition engine matches against an entry within the grammar. Recognition engines tend to be optimized to find the best possible match, which leads toward false acceptances instead of rejections. This second issue can be adjusted during site tuning, but the correction occurs long after application testing and acceptance is completed.

Voiyager allows the tester to provide the explicit input phrase, spoken or entered, by a caller and prevents input ambiguity. Testers can also indicate explicit **No Input** and **No Match** actions. For VoiceXML record and transfer actions, Voiyager allows the tester to provide a rich set of responses to the application so that all scenarios are easily tested.

Additionally, Voiyager provides a set of valid inputs in a drop-down menu making it easier to navigate the application by the tester.



## Speed

VoIP and telephone calls typically require some amount of initial setup time at the beginning of each playback to activate the recognizer, followed by a latency period that includes end-of-speech detection and speech recognition.

Voiyager Browser Phone calls are not limited by recognition engine initialization and recognition latency. Testers provide input as soon as the VoiceXML application provides the page and resources.

## Repeatability

Voiyager provides an additional advantage over traditional VoIP and telephony calls. Save or export Voiyager Browser Phone calls as Voiyager bookmarks to play back and execute the call testing again running nearly as fast as the application server provides the pages and resources for Voiyager to process. Five to ten dialog states are typically processed in 5-10 seconds.

Further, e-mail or attach Voiyager bookmarks to issues using your own issue tracking system so that developers can repeat problems discovered by testers in the development environment until the call problem is resolved.

## Purchasing and Licensing

### *How do I deploy Voiyager ?*

Provided as as a rack-mounted appliance server, Voiyager uses a Web server built into the solution. Download the Voiyager client software from the home page on the Voiyager server.

### *How is Voiyager licensed?*

Voiyager uses a per appliance server license plan.

### *How do I determine the number of Voiyager servers that I need?*

Voiyager supports multiple concurrent projects and is typically only limited by the number and size of the explorations. The typical expectation is that a Voiyager server supports two to three medium-sized concurrent projects including developers, testers, and analysts. Larger projects can require one or more servers each, depending on the number of teams with explorations that must run concurrently.

### Sales and Product Information

info@voiyager.com \ 1.877.800.0793

### Voiyager

16610 North Black Canyon Highway, Suite 100  
Phoenix, AZ 85053

