



**Cisco CVP VoiceXML 3.0**

---

**Element Specifications**

Publication date: 14 January 2005

Copyright (C) 2000-2005 Audium Corporation. All rights reserved.

Distributed by Cisco Systems, Inc. under license from Audium Corporation.  
Customer Voice Portal is a trademark of Cisco Systems, Inc.

### **Contact Information**

For more information, contact Cisco Systems, Inc. at the following location:

Corporate Headquarters  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706

Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 526-4100

General product information, visit <http://www.cisco.com>

Support inquiries, visit <http://www.cisco.com/support>

Customer forum, visit <http://www.cisco.com/discuss/contactcenter>

---

**ABOUT THIS DOCUMENT..... 8**

    AUDIENCE ..... 8

    RELATED DOCUMENTATION ..... 8

**OBTAINING TECHNICAL ASSISTANCE..... 9**

    CISCO.COM ..... 9

    NETWORK PROFESSIONAL CONNECTION ..... 9

    TECHNICAL ASSISTANCE CENTER ..... 10

*Cisco TAC Website*..... 10

*Cisco TAC Escalation Center* ..... 11

**OBTAINING ADDITIONAL PUBLICATIONS AND INFORMATION..... 12**

**INTRODUCTION ..... 13**

**AUDIO..... 15**

    AUDIO GROUPS ..... 15

*Audio Playback* ..... 15

**COUNTER ..... 16**

    SETTINGS ..... 16

    ELEMENT DATA ..... 16

    EXIT STATES ..... 16

**CURRENCY ..... 17**

    SETTINGS ..... 18

    ELEMENT DATA ..... 18

    EXIT STATES ..... 18

    AUDIO GROUPS ..... 19

*Currency Capture* ..... 19

*End* ..... 19

**CURRENCY\_WITH\_CONFIRM..... 20**

    SETTINGS ..... 21

    ELEMENT DATA ..... 21

    EXIT STATES ..... 22

    AUDIO GROUPS ..... 22

*Currency Capture* ..... 22

*Currency Confirm* ..... 22

*End* ..... 23

**DATABASE ..... 24**

    SETTINGS ..... 24

---

---

ELEMENT DATA.....	25
SESSION DATA.....	25
EXIT STATES.....	25
<b>DATE.....</b>	<b>26</b>
SETTINGS.....	26
ELEMENT DATA.....	26
EXIT STATES.....	27
AUDIO GROUPS .....	27
<i>Date Capture</i> .....	27
<i>End</i> .....	27
<b>DATE_WITH_CONFIRM .....</b>	<b>28</b>
SETTINGS.....	28
ELEMENT DATA.....	29
EXIT STATES.....	29
AUDIO GROUPS .....	29
<i>Date Capture</i> .....	29
<i>Date Confirm</i> .....	30
<i>End</i> .....	30
<b>DIGITS .....</b>	<b>31</b>
SETTINGS.....	31
ELEMENT DATA.....	32
EXIT STATES.....	32
AUDIO GROUPS .....	32
<i>Digits Capture</i> .....	32
<i>End</i> .....	32
<b>DIGITS_WITH_CONFIRM .....</b>	<b>34</b>
SETTINGS.....	34
ELEMENT DATA.....	35
EXIT STATES.....	35
AUDIO GROUPS .....	36
<i>Digits Capture</i> .....	36
<i>Digits Confirm</i> .....	36
<i>End</i> .....	37
<b>EMAIL .....</b>	<b>38</b>
SETTINGS.....	38
EXIT STATES.....	39
<b>FORM.....</b>	<b>40</b>
SETTINGS.....	40

---

---

ELEMENT DATA.....	44
EXIT STATES.....	45
AUDIO GROUPS .....	46
<i>Form Data Capture</i> .....	46
<i>End</i> .....	46
<b>FORM_WITH_CONFIRM.....</b>	<b>47</b>
SETTINGS.....	47
ELEMENT DATA.....	52
EXIT STATES.....	53
AUDIO GROUPS .....	53
<i>Form Data Capture</i> .....	53
<i>Form Data Confirm</i> .....	53
<i>End</i> .....	54
<b>2_OPTION_MENU, 3_OPTION_MENU, . . . , 10_OPTION_MENU.....</b>	<b>55</b>
SETTINGS.....	55
ELEMENT DATA.....	56
EXIT STATES.....	57
AUDIO GROUPS .....	57
<i>Menu Option Capture</i> .....	57
<i>End</i> .....	57
<b>NUMBER .....</b>	<b>59</b>
SETTINGS.....	59
ELEMENT DATA.....	59
EXIT STATES.....	60
AUDIO GROUPS .....	60
<i>Number Capture</i> .....	60
<i>End</i> .....	60
<b>NUMBER_WITH_CONFIRM.....</b>	<b>62</b>
SETTINGS.....	62
ELEMENT DATA.....	63
EXIT STATES.....	63
AUDIO GROUPS .....	63
<i>Number Capture</i> .....	63
<i>Number Confirm</i> .....	64
<i>End</i> .....	64
<b>PHONE.....</b>	<b>65</b>
SETTINGS.....	65
ELEMENT DATA.....	65

---

---

EXIT STATES.....	66
AUDIO GROUPS .....	66
<i>Phone Capture</i> .....	66
<i>End</i> .....	66
<b>PHONE_WITH_CONFIRM .....</b>	<b>67</b>
SETTINGS.....	67
ELEMENT DATA.....	68
EXIT STATES.....	68
AUDIO GROUPS .....	68
<i>Phone Capture</i> .....	68
<i>Phone Confirm</i> .....	69
<i>End</i> .....	69
<b>RECORD.....</b>	<b>70</b>
SETTINGS.....	70
ELEMENT DATA.....	72
EXIT STATES.....	72
AUDIO GROUPS .....	72
<i>Record Capture</i> .....	72
<b>RECORD_WITH_CONFIRM.....</b>	<b>73</b>
SETTINGS.....	73
ELEMENT DATA.....	75
EXIT STATES.....	75
AUDIO GROUPS .....	75
<i>Record Capture</i> .....	75
<i>Record Confirm</i> .....	76
<b>SUBDIALOG_RETURN .....</b>	<b>77</b>
SETTINGS.....	77
EXIT STATES.....	77
<b>SUBDIALOG_START .....</b>	<b>79</b>
SETTINGS.....	79
EXIT STATES.....	80
<b>TIME .....</b>	<b>81</b>
SETTINGS.....	81
ELEMENT DATA.....	81
EXIT STATES.....	82
AUDIO GROUPS .....	82
<i>Time Capture</i> .....	82
<i>End</i> .....	82

---

---

<b>TIME_WITH_CONFIRM.....</b>	<b>83</b>
SETTINGS.....	83
ELEMENT DATA.....	84
EXIT STATES.....	84
AUDIO GROUPS .....	84
<i>Time Capture</i> .....	84
<i>Time Confirm</i> .....	85
<i>End</i> .....	85
<b>TRANSFER.....</b>	<b>86</b>
SETTINGS.....	86
ELEMENT DATA.....	87
EXIT STATES.....	87
AUDIO GROUPS .....	88
<i>Transfer Audio</i> .....	88
<i>End</i> .....	88
<b>YES_NO_MENU .....</b>	<b>89</b>
SETTINGS.....	89
ELEMENT DATA.....	89
EXIT STATES.....	90
AUDIO GROUPS .....	90
<i>Yes / No Capture</i> .....	90
<i>End</i> .....	90

## About This Document

Thank you for choosing Cisco CVP VoiceXML 3.0. This document describes how use and configure CVP VoiceXML elements in your voice application.

### Audience

This document is intended for voice application and component developers using Cisco CVP VoiceXML.

### Related Documentation

- **Cisco CVP VoiceXML Studio.** The Cisco CVP VoiceXML Studio documentation describes the functionality of Studio including creating projects, using the Studio environment and deploying applications to CVP VoiceXML Server. This documentation is available through CVP VoiceXML Studio only.
- **Installation Guide.** This document provides instructions on installing Cisco CVP VoiceXML and its components.
- **User Guide.** This document provides an introduction to Cisco CVP VoiceXML software though it focuses on describing all aspects of CVP VoiceXML Server.
- **Programmer Guide.** This document provides information on how to build components that run on Cisco CVP VoiceXML.
- **Say It Smart Specifications.** This document describes in detail the functionality and configuration options for all CVP VoiceXML Say It Smart plugins included with the software.
- **Javadocs.** Javadocs are a group of HTML pages fully describing the entire Java application programming interfaces (APIs) to Cisco CVP VoiceXML. Developers use Javadocs in conjunction with the Programmer Guide to understand how to build custom components that run on Cisco CVP VoiceXML.

## Obtaining Technical Assistance

Cisco provides Cisco.com, which includes the Cisco Technical Assistance Center (TAC) Website, as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from the Cisco TAC website. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC website, including TAC tools and utilities.

### Cisco.com

Cisco.com offers a suite of interactive, networked services that let you access Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com provides a broad range of features and services to help you with these tasks:

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

To obtain customized information and service, you can self-register on Cisco.com at this URL:

<http://www.cisco.com>

### Network Professional Connection

Cisco provides a forum where you can discuss and exchange information regarding call center issues. To access the the forum, go to the following Web site:

<http://www.cisco.com/discuss/contactcenter>

---

## Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two levels of support are available: the Cisco TAC website and the Cisco TAC Escalation Center. The avenue of support that you choose depends on the priority of the problem and the conditions stated in service contracts, when applicable.

We categorize Cisco TAC inquiries according to urgency:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

### *Cisco TAC Website*

You can use the Cisco TAC website to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC website, go to this URL:

<http://www.cisco.com/tac>

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC website. Some services on the Cisco TAC website require a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

<http://tools.cisco.com/RPF/register/register.do>

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC website, you can open a case online at this URL:

<http://www.cisco.com/en/US/support/index.html>

If you have Internet access, we recommend that you open P3 and P4 cases through the Cisco TAC website so that you can describe the situation in your own words and attach any necessary files.

#### *Cisco TAC Escalation Center*

The Cisco TAC Escalation Center addresses priority level 1 or priority level 2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

<http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml>

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, please have available your service agreement number and your product serial number.

---

## Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

The *Cisco Product Catalog* describes the networking products offered by Cisco Systems as well as ordering and customer support services. Access the *Cisco Product Catalog* at this URL:

[http://www.cisco.com/en/US/products/products\\_catalog\\_links\\_launch.html](http://www.cisco.com/en/US/products/products_catalog_links_launch.html)

Cisco Press publishes a wide range of networking publications. Cisco suggests these titles for new and experienced users: *Internetworking Terms and Acronyms Dictionary*, *Internetworking Technology Handbook*, *Internetworking Troubleshooting Guide*, and the *Internetworking Design Guide*. For current Cisco Press titles and other information, go to Cisco Press online at this URL:

<http://www.ciscopress.com>

*Packet* magazine is the Cisco monthly periodical that provides industry professionals with the latest information about the field of networking. You can access *Packet* magazine at this URL:

[http://www.cisco.com/en/US/about/ac123/ac114/about\\_cisco\\_packet\\_magazine.html](http://www.cisco.com/en/US/about/ac123/ac114/about_cisco_packet_magazine.html)

*iQ Magazine* is the Cisco monthly periodical that provides business leaders and decision makers with the latest information about the networking industry. You can access *iQ Magazine* at this URL:

[http://business.cisco.com/prod/tree.taf%3fasset\\_id=44699&public\\_view=true&kbns=1.html](http://business.cisco.com/prod/tree.taf%3fasset_id=44699&public_view=true&kbns=1.html)

*Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in the design, development, and operation of public and private internets and intranets. You can access the *Internet Protocol Journal* at this URL:

[http://www.cisco.com/en/US/about/ac123/ac147/about\\_cisco\\_the\\_internet\\_protocol\\_journal.html](http://www.cisco.com/en/US/about/ac123/ac147/about_cisco_the_internet_protocol_journal.html)

Training—Cisco offers world-class networking training, with current offerings in network training listed at this URL:

[http://www.cisco.com/en/US/learning/le31/learning\\_recommended\\_training\\_list.html](http://www.cisco.com/en/US/learning/le31/learning_recommended_training_list.html)

---

# Introduction

Every element included with CVP VoiceXML Studio and Server must be configured before it can be used. This reference file contains a detailed specification for each of the CVP VoiceXML elements, listing all the options available in the configuration. The specifications must be followed, or the element may complain with an error message or behave erratically.

Each element specification in this reference file presents information on some or all of the following five topics:

- **Overview.** Each specification starts with a brief description of the element's behavior including what it does, how it reacts to various settings and audio groups, and other miscellaneous behavior. The information should help the developer decide whether to use these elements in an application or to rely on custom elements.
- **Settings.** Settings contain information that affects how the element behaves. Each setting is described in this document using the following attributes:
  - **Type:** The type of data accepted such as a boolean, integer, or enumeration.
  - **Required:** This defines whether the setting is required to have a value **if the setting is active** (available to be configured in CVP VoiceXML Studio). The definition of required in this case is that the setting must have an appropriate value for CVP VoiceXML Studio to validate the voice element configuration. In CVP VoiceXML Studio, all required settings are denoted by a red asterisk in front of the setting names.
  - **Single setting value:** This defines whether the setting can have multiple values. If set to true, then the setting may have only a single configuration value. Multiple value settings are created in CVP VoiceXML Studio by right clicking on the setting and choosing the "add *setting name*" option.
  - **Substitution allowed:** This setting attribute determines if the setting value can include substitution for dynamic configuration. In CVP VoiceXML Studio, any setting that allows substitution values will activate the substitution button located at the top right of the settings pane.
  - **Default:** The initial value of an element setting when a new element is dragged to the workspace.
- **Element Data.** Some elements capture data, and yield information that may be useful to other elements or for logging purposes. The variables created by each element are listed here.
- **Exit States.** Each element may have one or more exit states that indicate the dialog status when the element execution has completed. Exit states do not appear in an element configuration and cannot be changed.

- 
- **Audio Groups.** Audio groups define the different places within an element where audio can be played. Each audio group may contain one or more audio items that can be played one after another. Bargein is specified on the audio group level, and therefore all audio items within the same audio group will share the same bargein value. Audio items may be pre-recorded audio files, text-to-speech (TTS) phrases, and Say It Smart types (playback of formatted data such as dates, currency amounts, etc). Each audio group is described using the following attributes:
    - **Required:** This defines whether the audio group is required to have a configuration value, in order for CVP VoiceXML Studio to validate the voice element configuration. In CVP VoiceXML Studio, all required audio groups will show up in the Audio pane when a new element is dragged to the workspace, and they are not removable. Audio groups that are optional will be included in the “Add Audio Groups” list and can be configured after they are added to the Audio pane.
    - **Max 1:** This defines whether multiple audio groups can be configured. If set to Yes, the element may only have one such audio group configured, though the audio group may still contain more than one audio items. If set to No, multiple audio groups can be configured in CVP VoiceXML Studio by right clicking on Audio Groups (at top of the pane) and choosing the “add *audio group name*” option. In CVP VoiceXML Studio, any audio group that is limited to one configuration per element will not display a count in its name, and the Count drop-down box will be greyed out. Audio groups with multiple counts are used to define different audios to play each time a certain VoiceXML event occurs (often known as tapered prompts).

---

## Audio

The Audio voice element simply outputs a VoiceXML page with the contents of a single audio group. The Audio element is used for greetings, error messages and any other time audio is to be played in a situation not associated with an input state.

### Audio Groups

#### *Audio Playback*

Name (Label)	Max 1	Req'd	Notes
<code>initial_audio_group</code> (Initial)	Yes	Yes	The audio group containing the audio to play.

**Studio Element Folder:** Top Level

**Class Name:** `com.audium.server.voiceElement.audio.MAudio`

## Counter

The Counter action element is used to keep track of a count stored as element data. The initial value of the count is defined as a configuration setting. In addition, the element may be configured to increment or decrement with a user defined step size. A typical use for the Counter element would be in a loop in the call flow that increments the count until a decision element decides that the loop must end. Revisiting a Counter element instance will automatically update the count.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>initial</b> (Initial Count)	int	Yes	true	true	0	This setting specifies at which integer value this counter should start.
<b>type</b> (Type)	string enum	Yes	true	true	increment	This setting specifies whether the counter should be incremented or decremented. Possible values are: <code>decrement</code>   <code>increment</code> .
<b>step</b> (Step Size)	int	Yes	true	true	1	This setting specifies by how much this counter should be incremented or decremented.

### Element Data

Name	Type	Notes
<b>count</b>	String	The current count.

### Exit States

Name	Notes
<b>done</b>	The counter was updated.

**Studio Element Folder:** Calculation

**Class Name:** `com.audium.server.action.counter.CounterAction`

## Currency

The Currency voice element captures from the caller a currency amount in dollars and cents. The currency amount can be entered using the keypad or speech input. The captured value will be stored in element data as a decimal value (without the \$ character).

There are several different formats for speaking a currency amount or entering it through the keypad. Voice browsers may use different grammars and therefore accept different utterances. However, the spoken formats listed below should result in the same behavior for all supported browsers. The tables below list each input and the value that is stored in the element variable as a result. If some data is left out, the system assumes a default value for the missing information.

Utterance	Stored Value	Example	Description
[dollar] "dollar(s)" ("and") [cent] "cent(s)"	D.CC	"thirteen dollars and fifty cents " = 13.50	Dollars are whole numbers >= 0. Cents are from 00 to 99. The word "and" is optional.
[dollar] "dollar(s) "[cent]	D.CC	"thirteen dollars five" = 13.05	Dollars are whole numbers >= 0. Cents are from 00 to 99.
[dollar] "dollar(s)"	D.00	"three hundred fifty" = 350.00	A plain whole number is interpreted as dollars with no cents.
[cent] "cent(s)"	0.CC	"three cents" = 0.03	To specify cents only, the word "cents" but be uttered. Cents are from 00 to 99.

DTMF Entry	Stored Value	Example	Description
[D]*[CC]	D.CC	3*99 = 3.99	The decimal is represented by the * button.

There are other formats that are possible, particularly when entering via DTMF and inputting incomplete amounts. These inputs may yield differing results on various voice browsers. The returned variable will always be a decimal value with the appropriate number of padded zeros if applicable.

## Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	string enum	Yes	true	false	both	The type of entry allowed for input. Possible values are: <code>voice</code>   <code>dtmf</code>   <code>both</code> .
<b>noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<b>currency_max_noinput_count</b> (Currency Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during currency input capture. 0 = infinite noinputs allowed.
<b>currency_max_nomatch_count</b> (Currency Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during currency input capture. 0 = infinite nomatches allowed.
<b>currency_confidence_level</b> (Currency Confidence Level)	decimal (0.0 to 1.0)	Yes	true	true	0.40	The confidence level threshold to use during currency capture.

## Element Data

Name	Type	Notes
<b>value</b>	string	The currency amount captured. This will always be a decimal number with the appropriate number of padded zeros (up to 2).
<b>value_confidence</b>	float	This is the confidence value of the captured utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>done</b>	The currency capture was completed.

## Audio Groups

### *Currency Capture*

Name (Label)	Req'd	Max 1	Notes
<b>currency_initial_audio_group</b> (Currency Initial)	Yes	Yes	Played when the voice element first begins.
<b>currency_nomatch_audio_group</b> (Currency NoMatch)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>currency_noinput_audio_group</b> (Currency NoInput)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>currency_help_audio_group</b> (Currency Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### *End*

Name (Label)	Req'd	Max 1	Notes
<b>done_audio_group</b> (Done)	No	Yes	Played when the currency capture is completed and the voice element exits with the <b>done</b> exit state.

**Studio Element Folder:** Commerce

**Class Name:** `com.audium.server.voiceElement.currency.MBasicCurrency`

## Currency\_With\_Confirm

The Currency\_With\_Confirm voice element captures from the caller a currency amount in dollars and cents, and presents a confirmation menu allowing the caller to either accept their entry or re-enter the currency value. The currency amount can be entered using the keypad or speech input. The captured value will be stored in element data as a decimal value (without the \$ character).

There are several different formats for speaking a currency amount or entering it through the keypad. Voice browsers may use different grammars and therefore accept different utterances. However, the spoken formats listed below should result in the same behavior for all supported browsers. The tables below list each input and the value that is stored in element data as a result. If some data is left out, the system assumes a default value for the missing information.

Utterance	Stored Value	Example	Description
[dollar] "dollar(s)" ("and") [cent] "cent(s)"	D.CC	"thirteen dollars and fifty cents " = 13.50	Dollars are whole numbers $\geq 0$ . Cents are from 00 to 99. The word "and" is optional.
[dollar] "dollar(s)" [cent]	D.CC	"thirteen dollars five" = 13.05	Dollars are whole numbers $\geq 0$ . Cents are from 00 to 99.
[dollar] "dollar(s)"	D.00	"three hundred fifty" = 350.00	A plain whole number is interpreted as dollars with no cents.
[cent] "cent(s)"	0.CC	"three cents" = 0.03	To specify cents only, the word "cents" but be uttered. Cents are from 00 to 99.

DTMF Entry	Stored Value	Example	Description
[D]*[CC]	D.CC	3*99 = 3.99	The decimal is represented by the * button.

There are other formats that are possible, particularly when entering via DTMF and inputting incomplete amounts. These inputs may yield inconsistent results on various voice browsers. The returned variable will always be a decimal value with the appropriate number of padded zeros if applicable.

## Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	string enum	Yes	true	false	both	The type of entry allowed for input. Possible values are: <code>voice</code>   <code>dtmf</code>   <code>both</code> .
<b>noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a <code>noinput</code> event is thrown.
<b>currency_max_noinput_count</b> (Currency Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of <code>noinput</code> events allowed during currency input capture. 0 = infinite <code>noinputs</code> allowed.
<b>currency_max_nomatch_count</b> (Currency Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of <code>nomatch</code> events allowed during currency input capture. 0 = infinite <code>nomatches</code> allowed.
<b>confirm_max_noinput_count</b> (Confirm Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of <code>noinput</code> events allowed during currency input confirmation. 0 = infinite <code>noinputs</code> allowed.
<b>confirm_max_nomatch_count</b> (Confirm Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of <code>nomatch</code> events allowed during currency input confirmation. 0 = infinite <code>nomatches</code> allowed.
<b>max_disconfirmed_count</b> (Max Disconfirmed Count)	int ≥ 0	Yes	true	true	3	The maximum number of times a caller is allowed to disconfirm a captured input. 0 = infinite <code>disconfirmations</code> allowed.
<b>currency_confidence_level</b> (Currency Confidence Level)	decimal (0.0 to 1.0)	Yes	true	true	0.40	The confidence level threshold to use during currency capture.
<b>confirm_confidence_level</b> (Confirm Confidence Level)	Decimal (0.0 to 1.0)	Yes	true	true	0.50	The confidence level threshold to use during confirmation.

## Element Data

Name	Type	Notes
<b>value</b>	string	The currency amount captured. This will always be a decimal number with the appropriate number of padded zeros (up to 2).
<b>value_confidence</b>	float	This is the confidence value of the captured currency utterance.
<b>confirm_confidence</b>	float	This is the confidence value of the captured confirm utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>max_disconfirmed</b>	The maximum number of disconfirmation has occurred. If the max disconfirmed count is set to 0, this exit state will never occur.
<b>done</b>	The currency captured was confirmed.

## Audio Groups

### *Currency Capture*

Name (Label)	Req'd	Max 1	Notes
<b>currency_initial_audio_group</b> (Currency Initial)	Yes	Yes	Played when the voice element first begins.
<b>currency_nomatch_audio_group</b> (Currency NoMatch)	No	No	Played when a nomatch event occurs during currency input capture. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>currency_noinput_audio_group</b> (Currency NoInput)	No	No	Played when a noinput event occurs during currency input capture. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>currency_help_audio_group</b> (Currency Help)	No	No	Played when a help event is triggered during currency input capture. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### *Currency Confirm*

Name (Label)	Req'd	Max 1	Notes
<b>confirm_initial_audio_group</b> (Confirm Initial)	Yes	Yes	Played when confirmation first begins.
<b>confirm_nomatch_audio_group</b> (Confirm NoMatch)	No	No	Played when a nomatch event occurs during confirmation. The nomatch event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a nomatch.
<b>confirm_noinput_audio_group</b> (Confirm NoInput)	No	No	Played when a noinput event occurs during confirmation. The noinput event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a noinput.

<b>confirm_help_audio_group</b> (Confirm Help)	No	No	Played when a help event is triggered during confirmation. The help event count corresponds to the audio group count. If not specified, a help event will trigger the confirm nomatch audio group to play (or the confirm initial audio group if confirm nomatch audio group is not specified).
<b>disconfirmed_audio_group</b> (Disconfirmed)	No	No	Played after the caller disconfirms a captured currency entry. Upon reaching the <b>max_disconfirmed_count</b> , the prompt should be about exiting with the <b>max_disconfirmed</b> exit state.

*End*

Name (Label)	Req'd	Max 1	Notes
<b>yes_audio_group</b> (Yes)	No	Yes	Played after the caller chooses the "yes" option. If not specified, no audio will be played when this option is chosen.

**Studio Element Folder:** Commerce**Class Name:** com.audium.server.voiceElement.currency.MBasicCurrencyWithConfirm

## Database

The database element provides the ability to execute an SQL command on external databases within a voice application call flow. The element requires JNDI to be configured in the Java application server to handle database connections. Only a single SQL statement can be executed per element. There are four types of commands that can be made:

- **Single.** This is used to run a SQL query that returns only a single row. Element data will be created with the variable names being the names of the columns returned and the value of that column as the element data value (as a string). If no row is returned, no element data will be set.
- **Multiple.** This is used to run a SQL query that returns multiple rows. A CVP-defined Java data structure, the Java class `ResultSetList`, stores the full result and is placed in session data. If no rows are returned, the `ResultSetList` object in session data will be empty. For detail about the `ResultSetList` data structure, refer to the Javadocs for this class.
- **Inserts.** This is used to run a SQL INSERT command that inserts information into the database.
- **Updates.** This is used to run a SQL UPDATE command that updates information in the database.

The developer can utilize substitution to create dynamic queries. The Database element is ideal for performing simple queries and updates. It may not be sufficient for performing complex database interactions such as multiple dependent queries or stored procedure calls. One would use a custom configurable or generic action element for these tasks. Also note that in order to avoid performance issues creating database connections, implementing database pooling on the application server is highly recommended.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>type</b> (Type)	string enum	Yes	true	true	single	The type of query: single, multiple, insert or update.
<b>jndiName</b> (JNDI Name)	string	Yes	true	true	None	This JNDI name for the SQL datasource of the database.
<b>key</b> (Session Data Key)	string	Yes	true	true	None	For queries of type multiple, the name of the session variable which the results of the query will be stored.
<b>query</b> (SQL Query)	string	Yes	true	true	None	The SQL query to be executed.

## Element Data

Element data is created *only* when the “type” setting is set to “single”. Element data given the names of the return columns are created containing the respective return values. For example, if a query returned the following information:

```
foo   bar
123   456
```

The following element data will be created: “foo” with the value “123” and “bar” with the value “456”.

## Session Data

Session data is created *only* when the “type” setting is set to “multiple”. In all other cases, no session data is created.

Name	Type	Notes
[value of setting “key”]	ResultSetList	The Java data structure that stores the returned values from a multiple type query. The name of the session data variable is specified by the developer in the “key” setting.

## Exit States

Name	Notes
done	The database query successfully completed.

**Studio Element Folder:** Integration

**Class Name:** com.audium.server.action.database.DatabaseAction

## Date

The Date voice element captures a date input from the caller. The date can be entered using DTMF input (in the YYYYMMDD format). It can also be spoken in natural language including a month, day and year. The captured value will be stored in element data as a fixed-length date string in the YYYYMMDD format. If the year is not specified in the input, YYYY is stored as “????”. And if the month or the day is not specified, MM and DD will be stored as “??”.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	string enum	Yes	true	false	<i>both</i>	The type of entry allowed for input. Possible values are: <i>voice</i>   <i>dtmf</i>   <i>both</i> .
<b>Noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	<i>5s</i>	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<b>collect_max_noinput_count</b> (Date Max NoInput Count)	int $\geq 0$	Yes	true	true	3	The maximum number of noinput events. 0 = infinite noinputs allowed.
<b>collect_max_nomatch_count</b> (Date Max NoMatch Count)	int $\geq 0$	Yes	true	false	3	The maximum number of nomatch events allowed. 0 = infinite nomatches allowed.
<b>collect_confidence_level</b> (Date Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use during date capture.

### Element Data

Name	Type	Notes
<b>value</b>	string	The date stored in the YYYYMMDD format.
<b>value_confidence</b>	float	This is the confidence value of the captured date utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the max nomatch count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the max noinput count is 0, this exit state will never occur.
<b>done</b>	The date capture was completed.

## Audio Groups

### *Date Capture*

Name (Label)	Req'd	Max 1	Notes
<b>collect_initial_audio_group</b> (Date Initial)	Yes	Yes	Played when the voice element first begins.
<b>collect_nomatch_audio_group</b> (Date NoMatch)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>collect_noinput_audio_group</b> (Date NoInput)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>collect_help_audio_group</b> (Date Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### *End*

Name (Label)	Req'd	Max 1	Notes
<b>done_audio_group</b> (Done)	No	Yes	Played after the date capture is completed. If not specified, no audio will be played.

### Studio Element Folder: Date & Time

**Class Name:** com.audium.server.voiceElement.date.MBasicDate

## Date\_With\_Confirm

The Date\_With\_Confirm voice element captures a date input from the caller, and presents a confirmation menu allowing the caller to either accept their entry or re-enter the date. The date can be entered using DTMF input (in the YYYYMMDD format). It can also be spoken in natural language including a month, day and year. The captured value will be stored in element data as a fixed-length date string in the YYYYMMDD format. If the year is not specified in the input, YYYY is stored as “????”. If the month or the day is not specified, MM and DD will be stored as “??”.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	string enum	Yes	true	false	<i>both</i>	The type of entry allowed for input. Possible values are: <i>voice</i>   <i>dtmf</i>   <i>both</i> .
<b>noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	<i>5s</i>	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<b>collect_max_noinput_count</b> (Date Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during date input capture. 0 = infinite noinputs allowed.
<b>collect_max_nomatch_count</b> (Date Max NoMatch Count)	int ≥ 0	Yes	true	false	3	The maximum number of nomatch events allowed during date input capture. 0 = infinite nomatches allowed.
<b>confirm_max_noinput_count</b> (Confirm Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during date input confirmation. 0 = infinite noinputs allowed.
<b>confirm_max_nomatch_count</b> (Confirm Max NoMatch Count)	int ≥ 0	Yes	true	false	3	The maximum number of nomatch events allowed during date input confirmation. 0 = infinite nomatches allowed.
<b>max_disconfirmed_count</b> (Max Disconfirmed Count)	int ≥ 0	Yes	true	false	3	The maximum number of times a caller is allowed to disconfirm a captured input. 0 = infinite disconfirmations allowed.
<b>collect_confidence_level</b> (Date Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use during date capture.
<b>confirm_confidence_level</b> (Confirm Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.50	The confidence level threshold to use during confirmation.

## Element Data

Name	Type	Notes
<b>value</b>	string	The date stored in the YYYYMMDD format.
<b>value_confidence</b>	float	This is the confidence value of the captured date utterance.
<b>confirm_confidence</b>	float	This is the confidence value of the captured confirm utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the max nomatch count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the max noinput count is 0, this exit state will never occur.
<b>max_disconfirmed</b>	The maximum number of disconfirmation occurred. If the <b>max_disconfirmed_count</b> is set to 0, this exit state will never occur.
<b>done</b>	The date captured was confirmed.

## Audio Groups

### *Date Capture*

Name (Label)	Req'd	Max 1	Notes
<b>collect_initial_audio_group</b> (Date Initial)	Yes	Yes	Played when the voice element first begins.
<b>collect_nomatch_audio_group</b> (Date NoMatch)	No	No	Played when a nomatch event occurs during date input capture. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>collect_noinput_audio_group</b> (Date NoInput)	No	No	Played when a noinput event occurs during date input capture. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>collect_help_audio_group</b> (Date Help)	No	No	Played when a help event is triggered during date input capture. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

*Date Confirm*

Name (Label)	Req'd	Max 1	Notes
<b>confirm_initial_audio_group</b> (Confirm Initial)	Yes	Yes	Played when the captured date is confirmed.
<b>confirm_nomatch_audio_group</b> (Confirm NoMatch)	No	No	Played when a nomatch event occurs during confirmation. The nomatch event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a nomatch.
<b>confirm_noinput_audio_group</b> (Confirm NoInput)	No	No	Played when a noinput event occurs during confirmation. The noinput event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a noinput.
<b>confirm_help_audio_group</b> (Confirm Help)	No	No	Played when a help event is triggered during confirmation. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the confirm nomatch audio group to play (or the confirm initial audio group if confirm nomatch audio group is not specified).
<b>disconfirmed_audio_group</b> (Disconfirmed)	No	No	Played after the caller disconfirms a date entry.

*End*

Name (Label)	Req'd	Max 1	Notes
<b>yes_audio_group</b> (Yes)	No	Yes	Played after the caller chooses the "yes" option. If not specified, no audio will be played when this option is chosen.

**Studio Element Folder:** Date & Time**Class Name:** com.audium.server.voiceElement.date.MBasicDateWithConfirm

## Digits

The Digits voice element captures a string of numerical digits. It may be used to collect small or large strings of digits. The digit string can be entered using the keypad or speech input. The captured value will be stored in element data as a string. The string cannot contain any non-numerical characters. Using speech input, the number is spoken one digit at a time (i.e. 49678 is spoken "four nine six seven eight"). DTMF input can be terminated by a # keypress if desired (if not used, the entry is considered terminated when the input timeout has been reached).

With the Digits voice element, the application designer has the ability to set length restrictions on the digit string. A minimum and maximum length can be given to narrow the criteria. If a string of a specific length is required, the minimum and maximum lengths should be set to the same value. If fewer digits are entered, a nomatch event will be thrown. A string of digits with length greater than the maximum length cannot be entered.

## Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	string enum	Yes	true	false	both	The type of entry allowed for input. Possible values are: voice   dtmf   both.
<b>noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<b>max_noinput_count</b> (Digits Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during digits input capture. 0 = infinite noinputs allowed.
<b>max_nomatch_count</b> (Digits Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during digits input capture. 0 = infinite nomatches allowed.
<b>digits_confidence_level</b> (Digits Confidence Level)	decimal (0.0 to 1.0)	Yes	true	true	0.40	The confidence level threshold to use during digits capture.
<b>min_digit</b> (Min Digit)	int > 0	Yes	true	true	None	Minimum number of digits allowed.
<b>max_digit</b> (Max Digit)	int > 0	Yes	true	true	None	Maximum number of digits allowed.

## Element Data

Name	Type	Notes
<code>value</code>	string	The digit string value captured.
<code>value_confidence</code>	float	This is the confidence value of the captured utterance.

## Exit States

Name	Notes
<code>max_nomatch</code>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<code>max_noinput</code>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<code>done</code>	The digit string capture was completed.

## Audio Groups

### *Digits Capture*

Name (Label)	Req'd	Max 1	Notes
<code>digits_initial_audio_group</code> (Digits Initial)	Yes	Yes	Played when the voice element first begins.
<code>digits_nomatch_audio_group</code> (Digits NoMatch)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<code>digits_noinput_audio_group</code> (Digits NoInput)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<code>digits_help_audio_group</code> (Digits Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### *End*

Name (Label)	Req'd	Max 1	Notes
<code>done_audio_group</code> (Done)	No	Yes	Played when the digits capture is completed and the voice element exits with the <code>done</code> exit state.

**Studio Element Folder:** Number Capture

**Class Name:** `com.audium.server.voiceElement.digit.MBasicDigit`

## Digits\_With\_Confirm

The Digits\_With\_Confirm voice element captures a string of numerical digits, and presents a confirmation menu allowing the caller to either accept their entry or re-enter the digits. It may be used to collect small or large strings of digits. The digit string can be entered using the keypad or speech input. The captured value will be stored in element data as a string. The string cannot contain non-numerical characters. Using speech input, the number is spoken one digit at a time (i.e. 49678 is spoken "four nine six seven eight"). DTMF input can be terminated by a # keypress if desired (otherwise, the entry is considered terminated when the input timeout is reached).

With the Digits\_With\_Confirm voice element, the application designer has the ability to set length restrictions on the digit string. A minimum and maximum length can be given to narrow the criteria. If a string of a specific length is required, the minimum and maximum lengths should be set to the same value. If fewer digits are entered, a nomatch event will be thrown. A string of digits with length greater than the maximum length cannot be entered.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<code>inputmode</code> (Input Mode)	string enum	Yes	true	false	both	The type of entry allowed for input (during digits capture and confirmation). Possible values are: <code>voice   dtmf   both</code> .
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<code>digits_max_noinput_count</code> (Digits Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during digits input capture. 0 = infinite noinputs allowed.
<code>digits_max_nomatch_count</code> (Digits Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during digits input capture. 0 = infinite nomatches allowed.
<code>confirm_max_noinput_count</code> (Confirm Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during digits input confirmation. 0 = infinite noinputs allowed.
<code>confirm_max_nomatch_count</code> (Confirm Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during digits input confirmation. 0 = infinite

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
						nomatches allowed.
<b>max_disconfirmed_count</b> (Max Disconfirmed Count)	int ≥ 0	Yes	true	true	3	The maximum number of times a caller is allowed to disconfirm a captured digits input. 0 = infinite disconfirmations allowed.
<b>digits_confidence_level</b> (Digits Confidence Level)	decimal (0.0 to 1.0)	Yes	true	true	0.40	The confidence level threshold to use during digits capture.
<b>confirm_confidence_level</b> (Confirm Confidence Level)	decimal (0.0 to 1.0)	Yes	true	true	0.50	The confidence level threshold to use during confirmation.
<b>min_digit</b> (Min Digit)	int > 0	Yes	true	true	None	Minimum number of digits allowed.
<b>max_digit</b> (Max Digit)	int > 0	Yes	true	true	None	Maximum number of digits allowed.

## Element Data

Name	Type	Notes
<b>value</b>	string	The digit string captured.
<b>value_confidence</b>	float	This is the confidence value of the captured digit string utterance.
<b>confirm_confidence</b>	float	This is the confidence value of the captured confirm utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>max_disconfirmed</b>	The maximum number of disconfirmation has occurred. If the max disconfirmed count is set to 0, this exit state will never occur.
<b>done</b>	The digit string captured was confirmed.

## Audio Groups

### *Digits Capture*

Name (Label)	Req'd	Max 1	Notes
<b>digits_initial_audio_group</b> ( <b>Digits Initial</b> )	Yes	Yes	Played when the voice element first begins.
<b>digits_nomatch_audio_group</b> ( <b>Digits NoMatch</b> )	No	No	Played when a nomatch event occurs during digits input capture. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>digits_noinput_audio_group</b> ( <b>Digits NoInput</b> )	No	No	Played when a noinput event occurs during digits input capture. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>digits_help_audio_group</b> ( <b>Digits Help</b> )	No	No	Played when a help event is triggered during digits input capture. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### *Digits Confirm*

Name (Label)	Req'd	Max 1	Notes
<b>confirm_initial_audio_group</b> ( <b>Confirm Initial</b> )	Yes	Yes	Played when confirmation first begins.
<b>confirm_nomatch_audio_group</b> ( <b>Confirm NoMatch</b> )	No	No	Played when a nomatch event occurs during confirmation. The nomatch event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a nomatch.
<b>confirm_noinput_audio_group</b> ( <b>Confirm NoInput</b> )	No	No	Played when a noinput event occurs during confirmation. The noinput event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a noinput.
<b>confirm_help_audio_group</b> ( <b>Confirm Help</b> )	No	No	Played when a help event is triggered during confirmation. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the confirm nomatch audio group to play (or the confirm initial audio group if confirm nomatch audio group is not specified).
<b>disconfirmed_audio_group</b> ( <b>Disconfirmed</b> )	No	No	Played after the caller disconfirms a captured digits entry. Upon reaching the <b>max_disconfirmed_count</b> , the prompt should be about exiting with the

Name (Label)	Req'd	Max 1	Notes
			<b>max_disconfirmed</b> exit state.

*End*

Name (Label)	Req'd	Max 1	Notes
<b>yes_audio_group</b> (Yes)	No	Yes	Played after the caller chooses the "yes" option. If not specified, no audio will be played when this option is chosen.

**Studio Element Folder:** Number Capture

**Class Name:** `com.audium.server.voiceElement.digit.MBasicDigitWithConfirm`

## Email

The Email action element sends messages using the Javamail package supplied by the application server to send messages to the provided email address. Additionally the message can include attachments. The application server must be configured to set a JNDI datasource for mail sessions. The *to* and *toList* fields are not individually required however at least one must be defined. Email addresses are not verified for syntax or validity. Attachments that do not exist will be skipped but the message will still be sent. Repeated email addresses are sent the message multiple times. The *toList*, *ccList* and *bccList* settings must refer to session data variables that holds a `ResultSetList` Java class holding a list of email addresses (retrieved from a Database element).

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>jndiName</b> (JNDI Name)	string	Yes	true	true	None	The configured JNDI datasource for mail sessions under the java application server.
<b>to</b> (To)	string	No	false	true	None	Email address this message should be sent to. This setting is repeatable so that each setting value contains a separate email address.
<b>toList</b> (To List)	string	No	true	true	None	The name of a session data variable containing a <code>ResultSetList</code> object holding a list of email addresses as retrieved from a Database element. The email will be sent to every address in this list.
<b>from</b> (From)	string	Yes	true	true	None	The from/reply-to address for the email.
<b>cc</b> (Cc)	string	No	false	true	None	Email address this message should be carbon copied to. This setting is repeatable so that each setting value contains a separate email address.
<b>ccList</b> (Cc List)	string	No	true	true	None	The name of a session data variable containing a <code>ResultSetList</code> object holding a list of email addresses as retrieved from a Database element. The email will be carbon copied to each address in this list.
<b>bcc</b> (Bcc)	string	No	false	true	None	Email address this message should be blind carbon copied to. This setting is repeatable so that each setting value contains a separate email address.
<b>bccList</b> (Bcc List)	string	No	true	true	None	The name of a session data variable containing a <code>ResultSetList</code> object holding a list of email addresses as retrieved from a Database element. The email will be blind carbon copied to each address in this list.
<b>subject</b> (Subject)	string	No	true	true	None	Subject field of the email.
<b>attachment</b> (Attachment)	string	No	false	true	None	Full local path of the file to be attached. This setting is repeatable so that each setting value contains a reference to separate attachments.

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>messageBody</b> (Message Body)	string	Yes	true	true	None	The message body of the email.

## Exit States

Name	Notes
<b>done</b>	The database query successfully completed.

**Studio Element Folder:** Notification

**Class Name:** `com.audium.server.action.email.EmailAction`

## Form

The Form voice element is used to capture any input from the caller, based on developer-specified grammars. The valid caller inputs can be specified either directly in the voice element settings (which will create an inline grammar) or with external grammar files. Information returned by the Form grammars are saved in element data that can then be analyzed by developer-defined components. A Form voice element can be configured to listen for voice input only, DTMF input only, or both voice and DTMF input. In short, the Form element is the most flexible of all CVP VoiceXML elements as it allows almost any custom information to be captured without requiring a separate voice element. If a CVP VoiceXML voice element does not capture certain information as desired, one can always use a Form element first, before embarking on constructing a custom voice element.

The Form element provides support for custom control over the VoiceXML code generation. For example, the developer can decide what name to use for the VoiceXML field, whether or not to include a field-level slot attribute and how to name the slot attribute. The element also supports separate options for activating help prompts and the ability to set modality for Form.

Multiple DTMF and speech external grammars can be referenced within a single Form element, and the developer has the ability to specify grammar weights for speech grammars and set MIME types for both speech and DTMF grammars. Additionally, the Form element can be used to capture multiple slots, and the developer can specify for which slot(s) they want the recognition values stored as element data. Advanced n-best processing can be enabled, and standard n-best results are stored in element data and the activity log.

## Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<code>inputmode</code> (Input Mode)	string enum	Yes	true	false	both	The type of entry allowed for input. Possible values are: voice   dtmf   both.
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no dtmf entry before a noinput event is thrown.
<code>form_max_noinput_count</code> (Form Max NoInput)	int $\geq 0$	Yes	true	true	3	0 = infinite noinputs allowed.
<code>form_max_nomatch_count</code> (Form Max NoMatch)	int $\geq 0$	Yes	true	true	3	0 = infinite nomatches allowed.
<code>form_confidence_level</code> (Form Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use for data capture.

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>voice_grammar</b> (Voice Grammar)	string	*No	false	true	None	<p>Defines an external voice grammar for Form, in a string format delimited with semi-colons specifying three values in the following order:</p> <ol style="list-style-type: none"> <li>1) The grammar weight (optional)</li> <li>2) The grammar type (optional)</li> <li>3) URL of the grammar file (required)</li> </ol> <p>Note that if one of the two optional values (weight and type) is defined, the same number of delimiters (i.e. two semi-colons) must be used. For example:</p> <ul style="list-style-type: none"> <li>• 0.6;application/srgs+xml;http://www.vxml.com/mygrammar.grxml;</li> <li>• application/srgs+xml;http://www.vxml.com/mygrammar.grxml</li> <li>• 0.6;;http://www.vxml.com/mygrammar.grxml</li> <li>• http://www.vxml.com/mygrammar.grxml</li> </ul> <p>This setting is repeatable so multiple external grammar sources may be specified. None of the four settings - <b>voice_grammar</b>, <b>dtmf_grammar</b>, <b>voice_keyword</b> and <b>dtmf_keypress</b>, are required, but at least one must be specified since a form cannot be completed without at least one grammar.</p>
<b>dtmf_grammar</b> (DTMF Grammar)	URI	*No	false	true	None	<p>Defines an external DTMF grammar for Form, in a string format delimited with a semi-colon specifying two values in the following order:</p> <ol style="list-style-type: none"> <li>1) The grammar type (optional)</li> <li>2) URL of the grammar file (required)</li> </ol> <p>Sample configuration values are:</p> <ul style="list-style-type: none"> <li>• application/srgs+xml;http://www.vxml.com/mygrammar.grxml</li> <li>• http://www.vxml.com/mygrammar.grxml</li> </ul> <p>This setting is repeatable so multiple external grammar sources may be specified. None of the four settings - <b>voice_grammar</b>, <b>dtmf_grammar</b>, <b>voice_keyword</b> and <b>dtmf_keypress</b>, are required, but at least one must be specified since a form cannot be completed without at least one grammar.</p>

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>voice_keyword</b> (Voice Keyword)	string	*No	false	true	None	<p>Defines the inline voice grammar for Form, with each configuration of this repeatable setting specifying one option for the grammar. The valid format is a string separated with a semi-colon specifying two values in the following order:</p> <ol style="list-style-type: none"> <li>1) The weight of the grammar item (optional)</li> <li>2) The grammar item (required)</li> </ol> <p>Note that the grammar item may either contain the input itself followed by an optional return value, or just the input. Sample configurations values are:</p> <ul style="list-style-type: none"> <li>• 0.6; news report [news]</li> <li>• 0.6; news report</li> <li>• news report [news]</li> <li>• news report</li> </ul> <p>None of the four settings - <b>voice_grammar</b>, <b>dtmf_grammar</b>, <b>voice_keyword</b> and <b>dtmf_keypress</b>, are required, but at least one must be specified since a form cannot be completed without at least one grammar.</p>
<b>dtmf_keypress</b> (DTMF Keypress)	character (0-9, #, *)	*No	false	true	None	<p>Defines the inline DTMF grammar for Form, with each configuration of this repeatable setting specifying one option for the grammar. The valid format is a character (0-9, #, *) representing the keypress, followed by an optional return value. Sample configurations values are:</p> <ul style="list-style-type: none"> <li>• 1 [news]</li> <li>• 1</li> </ul> <p>None of the four settings - <b>voice_grammar</b>, <b>dtmf_grammar</b>, <b>voice_keyword</b> and <b>dtmf_keypress</b>, are required, but at least one must be specified since a form cannot be completed without at least one grammar.</p>

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>help_voice_keyword</b> (Help Voice Keyword)	string	No	false	true	<i>None</i>	Specifies a custom inline voice grammar to activate the help audio group. Each value of this repeatable setting adds another valid utterance. The format is a string specifying just the utterance (e.g. "help please").  If this setting is configured, an inline help voice grammar will be generated, which will be active only within the current Form element.
<b>help_dtmf_keypress</b> (Help DTMF Keypress)	character (0-9, #, *)	No	false	true	<i>None</i>	Specifies a custom inline DTMF grammar to activate the help audio group. Each value of this repeatable setting adds another valid DTMF keypress. The format is a character (0-9, #, *) representing just the keypress.  If this setting is configured, an inline help DTMF grammar will be generated, which will be active only within the current Form element.
<b>modal</b> (Modal)	boolean	No	true	true	<i>false</i>	Whether or not to temporarily disable active grammars specified outside the current Form element (such as those defined by hotlinks). If set to true, only the current Form grammars will be enabled for the duration of the element. Otherwise all active grammars will be enabled.
<b>field_name</b> (Field Name)	string	Yes	true	true	<i>foundation_fld</i>	The value to assign to the VXML field name attribute.
<b>field_slot</b> (Field Slot)	string	No	true	true	<i>None</i>	The name to assign to the VXML field slot attribute. If left unspecified, the field will not include a slot attribute.
<b>slot_element_data</b> (Slot Element Data)	string	No	false	true	<i>None</i>	Specifies for which grammar slot the return value should be stored as element data. This is a repeatable setting so multiple slot names can be specified. See notes below for further details.
<b>maxnbest</b> (Maxnbest)	int $\geq$ 1	Yes	true	true	1	The maximum number of speech recognition results that can be generated per voice input.

## Notes:

- The voice browser requires the top-level slot name in a grammar (whether inline or external) to match the field-level slot attribute (or the field name attribute if the slot attribute is unspecified), in order for the field name variable (and hence the CVP VoiceXML “value” element data) to be defined. For inline grammars, the CVP VoiceXML Form element automatically generates the grammar slot name based on the slot attribute or the field name. For custom-defined grammars that are referenced from an external source, it is the developer’s job to configure “**Field Name**” and/or “**Field slot**” correctly so that the above requirement is met.
- For a grammar that returns either different slots for different inputs or multiple slots per utterance, there are two ways to configure the Form element to store the slot/value pair information:
  - Leave the “**slot\_element\_data**” setting empty. The Form element will create element data named “nbestInterpretationX” (where X is from 1 to the length of the n-best list) that contains a string that uses delimiters “+” and “:” to separate the multiple slot names from their values. For example: “+Slot1:value1+Slot2:value2...”. A developer would then need to parse this string in a subsequent element to obtain the different slot name and value pairs.
  - Configure the “**slot\_element\_data**” setting with the names for all the slots that can be returned. The Form element will create a new set of n-best element data to store the recognition results for each slot listed in that setting. The element data will be named as <SLOT\_ELEMENT\_DATA<X> (where “SLOT\_ELEMENT\_DATA” is a string identical to the setting value and X is from 1 to the length of the n-best list). For example, if “**slot\_element\_data**” have two values “city” and “state” and there are three n-best results triggered, then a total of six element data in the names of “city1”, “city2”, “city3”, “state1”, “state2”, and “state3” will be created to store each of the n-best values for the “city” and “state” slots. Note that if n-best processing is disabled by setting **maxnbest** to 1, then only one interpretation will be returned per input and thereby only one element data per slot (“city1” and “state1”) will be created.

## Element Data

Name	Type	Notes
<b>value</b>	string	This stores the value of the VXML field name variable.
<b>value_confidence</b>	float	This stores the confidence score of the captured Form utterance. When n-best recognition is enabled, this stores the confidence score of the top hypothesis in the n-best list.
<SLOT_ELEMENT_DATA1> <SLOT_ELEMENT_DATA2> ... <SLOT_ELEMENT_DATA* >	string	A separate set of element data stores the interpretation values for each filled slot of captured n-best utterances. While the maximum number of <SLOT_ELEMENT_DATA<X> values is equal to the <b>maxnbest</b> setting value, the actual number of these values available is dependent on speech recognition at runtime, where <SLOT_ELEMENT_DATA1> holds the slot value of the top hypothesis in the n-best list and <SLOT_ELEMENT_DATA<X> holds the slot value of the last hypothesis. Note that if the <b>slot_element_data</b>

Name	Type	Notes
		setting is blank, these sets of element data will <i>not</i> be created.
<b>nbestLength</b>	int ≥ 1	This stores the number of n-best hypotheses generated by the speech engine.
<b>nbestUtterance1</b> <b>nbestUtterance2</b> ... <b>nbestUtteranceX</b>	string	This set of element data stores the captured n-best utterances. While the maximum number of <b>nbestUtteranceX</b> values is equal to the <b>maxnbest</b> setting value, the actual number of these values available is determined by speech recognition at runtime, where <b>nbestUtterance1</b> holds the utterance of the top hypothesis in the n-best list and <b>nbestUtteranceX</b> holds the utterance of the last hypothesis.
<b>nbestInterpretation1</b> <b>nbestInterpretation2</b> ... <b>nbestInterpretationX</b>	string	This set of element data stores the interpretations of captured n-best utterances. While the maximum number of <b>nbestInterpretationX</b> values is equal to the <b>maxnbest</b> setting value, the actual number of these values available is determined by speech recognition at runtime, where <b>nbestInterpretation1</b> holds the interpretation of the top hypothesis in the n-best list and <b>nbestInterpretationX</b> holds the interpretation of the last hypothesis.
<b>nbestConfidence1</b> <b>nbestConfidence2</b> ... <b>nbestConfidenceX</b>	float	This set of element data stores the confidence scores of captured n-best utterances. While the maximum number of <b>nbestConfidenceX</b> values is equal to the <b>maxnbest</b> setting value, the actual number of these values available is determined by speech recognition at runtime, where <b>nbestConfidence1</b> holds the confidence score of the top hypothesis in the n-best list and <b>nbestConfidenceX</b> holds the confidence score of the last hypothesis.
<b>nbestInputmode1</b> <b>nbestInputmode2</b> ... <b>nbestInputmodeX</b>	string	This set of element data stores the input modes of captured n-best utterances.

\* “SLOT\_ELEMENT\_DATA” is a string identical to the configuration value of the “slot\_element\_data” setting, and X is from 1 to the length of the n-best list. If more than one such value is configured, then multiple sets of element data using the same naming convention will be created.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>done</b>	The caller input matched the grammar correctly.

## Audio Groups

### Form Data Capture

Name (Label)	Req'd	Max 1	Notes
<b>initial_audio_group</b> (Initial)	Yes	Yes	Played when the voice element first begins.
<b>nomatch_audio_group</b> (NoMatch)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>noinput_audio_group</b> (NoInput)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>help_audio_group</b> (Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### End

Name (Label)	Req'd	Max 1	Notes
<b>done_audio_group</b> (Done)	No	Yes	Played when the form data capture is completed, and the voice element exits with the <b>done</b> exit state.

### Studio Element Folder: Form

**Class Name:** com.audium.server.voiceElement.form.MFoundationForm

## Form\_With\_Confirm

The Form\_With\_Confirm voice element is used to capture and confirm input from the caller, based on application designer-specified grammars. The valid caller inputs can be specified either directly in the voice element settings (which will create an inline grammar) or with external grammar files. Information returned by the grammar are saved in element data that can then be analyzed by developer-defined components. A Form\_With\_Confirm voice element can be configured to listen for voice input only, DTMF input only, or both voice and DTMF input. In short, the Form\_With\_Confirm element is the most flexible of all CVP VoiceXML elements that have confirmation menus as it allows almost any custom information to be captured and confirmed without requiring a separate voice element. If a CVP VoiceXML voice element does not capture and confirm certain information as desired, one can always use a Form\_With\_Confirm element first, before embarking on constructing a custom voice element.

The Form\_With\_Confirm element provides support for custom control over the VoiceXML code generation. For example, the developer can decide what name to use for the VoiceXML field, whether or not to include a field-level slot attribute and how to name the slot attribute. The element also supports separate options for activating help prompts and the ability to set modality for Form.

Multiple DTMF and speech external grammars can be referenced within a single Form\_With\_Confirm element, and the developer has the ability to specify grammar weights for speech grammars and set MIME types for both speech and DTMF grammars. Additionally, the Form\_With\_Confirm element can be used to capture multiple slots, and the developer can specify for which slot(s) they want the recognition values stored as element data. Advanced n-best processing can be enabled, and standard n-best results are stored in element data and the activity log.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	string enum	Yes	true	false	both	The type of entry allowed for input. Possible values are: voice   dtmf   both.
<b>noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<b>form_max_noinput_count</b> (Form Max NoInput)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during form input capture. 0 = infinite noinputs allowed.
<b>form_max_nomatch_count</b> (Form Max NoMatch)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during form input capture. 0 = infinite nomatches allowed.

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>confirm_max_noinput_count</b> (Confirm Max NoInput)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during form input confirmation. 0 = infinite noinputs allowed.
<b>confirm_max_nomatch_count</b> (Confirm Max NoMatch)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during form input confirmation. 0 = infinite nomatches allowed.
<b>max_disconfirmed_count</b> (Max Disconfirmed Count)	int ≥ 0	Yes	true	true	3	The maximum number of times a caller is allowed to disconfirm a captured input. 0 = infinite disconfirmations allowed.
<b>form_confidence_level</b> (Form Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use for capture of the form data.
<b>confirm_confidence_level</b> (Confirm Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.50	The confidence level threshold to use for confirmation of the form data.
<b>voice_grammar</b> (Voice Grammar)	string	*No	false	true	None	<p>Defines an external voice grammar for Form_With_Confirm, in a string format delimited with semi-colons specifying three values in the following order:</p> <ol style="list-style-type: none"> <li>1) The grammar weight (optional)</li> <li>2) The grammar type (optional)</li> <li>3) URL of the grammar file (required)</li> </ol> <p>Note that if one of the two optional values (weight and type) is defined, the same number of delimiters (i.e. two semi-colons) must be used. For example:</p> <ul style="list-style-type: none"> <li>• 0.6;application/srgs+xml;http://www.vxml.com/mygrammar.grxml;</li> <li>• application/srgs+xml;http://www.vxml.com/mygrammar.grxml</li> <li>• 0.6;;http://www.vxml.com/mygrammar.grxml</li> <li>• http://www.vxml.com/mygrammar.grxml</li> </ul> <p>This setting is repeatable so multiple external grammar sources may be specified. None of the four settings - <b>voice_grammar</b>, <b>dtmf_grammar</b>, <b>voice_keyword</b> and <b>dtmf_keypress</b>, are required, but at least one must be specified since a form cannot be completed without at least one grammar.</p>

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>dtmf_grammar</b> (DTMF Grammar)	URI	*No	false	true	<i>None</i>	<p>Defines an external DTMF grammar for Form_With_Confirm, in a string format delimited with a semi-colon specifying two values in the following order:</p> <ol style="list-style-type: none"> <li>1) The grammar type (optional)</li> <li>2) URL of the grammar file (required)</li> </ol> <p>Sample configuration values are:</p> <ul style="list-style-type: none"> <li>• application/srgs+xml;http://www.vxml.com/mygrammar.xml</li> <li>• http://www.vxml.com/mygrammar.grxml</li> </ul> <p>This setting is repeatable so multiple external grammar sources may be specified. None of the four settings - <b>voice_grammar</b>, <b>dtmf_grammar</b>, <b>voice_keyword</b> and <b>dtmf_keypress</b>, are required, but at least one must be specified since a form cannot be completed without at least one grammar.</p>
<b>voice_keyword</b> (Voice Keyword)	string	*No	false	true	<i>None</i>	<p>Defines the inline voice grammar for Form_With_Confirm, with each configuration of this repeatable setting specifying one option for the grammar. The valid format is a string separated with a semi-colon specifying two values in the following order:</p> <ol style="list-style-type: none"> <li>1) The weight of the grammar item (optional)</li> <li>2) The grammar item (required)</li> </ol> <p>Note that the grammar item may either contain the input itself followed by an optional return value, or just the input. Sample configurations values are:</p> <ul style="list-style-type: none"> <li>• 0.6; news report [news]</li> <li>• 0.6; news report</li> <li>• news report [news]</li> <li>• news report</li> </ul> <p>None of the four settings - <b>voice_grammar</b>, <b>dtmf_grammar</b>, <b>voice_keyword</b> and <b>dtmf_keypress</b>, are required, but at least one must be specified since a form cannot be completed without at least one grammar.</p>

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>dtmf_keypress</b> (DTMF Keypress)	character (0-9, #, *)	*No	false	true	<i>None</i>	<p>Defines the inline DTMF grammar for Form_With_Confirm, with each configuration of this repeatable setting specifying one option for the grammar. The valid format is a character (0-9, #, *) representing the keypress, followed by an optional return value. Sample configurations values are:</p> <ul style="list-style-type: none"> <li>• 1 [news]</li> <li>• 1</li> </ul> <p>None of the four settings - <b>voice_grammar</b>, <b>dtmf_grammar</b>, <b>voice_keyword</b> and <b>dtmf_keypress</b>, are required, but at least one must be specified since a form cannot be completed without at least one grammar.</p>
<b>help_voice_keyword</b> (Help Voice Keyword)	string	No	false	true	<i>None</i>	<p>Specifies a custom inline voice grammar to activate the help audio group. Each value of this repeatable setting adds another valid utterance. The format is a string specifying just the utterance (e.g. "help please").</p> <p>If this setting is configured, an inline help voice grammar will be generated, which will be active only within the current Form With Confirm element.</p>
<b>help_dtmf_keypress</b> (Help DTMF Keypress)	character (0-9, #, *)	No	false	true	<i>None</i>	<p>Specifies a custom inline DTMF grammar to activate the help audio group. Each value of this repeatable setting adds another valid DTMF keypress. The format is a character (0-9, #, *) representing just the keypress.</p> <p>If this setting is configured, an inline help DTMF grammar will be generated, which will be active only within the current Form With Confirm element.</p>
<b>modal</b> (Modal)	boolean	No	true	true	<i>false</i>	<p>Whether or not to temporarily disable active grammars specified outside the current Form_With_Confirm element (such as those defined by hotlinks). If set to true, only the current Form_With_Confirm grammars will be enabled for the duration of the element. Otherwise all active grammars will be enabled.</p>
<b>field_name</b> (Field Name)	string	Yes	true	true	<i>foundation_fld</i>	The value to assign to the VXML field-level name attribute.

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>field_slot</b> (Field Slot)	string	No	true	true	None	The name to assign to the VXML field-level slot attribute. If left unspecified (i.e. the default value), the field will not have a slot attribute.
<b>slot_element_data</b> (Slot Element Data)	string	No	false	true	None	Specifies for which grammar slot the return value should be stored as element data. This is a repeatable setting so multiple slot names can be specified. See notes below for further details.
<b>maxnbest</b> (Maxnbest)	int ≥ 1	Yes	true	true	1	The maximum number of speech recognition results that can be generated per voice input.

## Notes:

- The voice browser requires the top-level slot name in a grammar (whether inline or external) to match the field-level slot attribute (or the field name attribute if the slot attribute is unspecified), in order for the field name variable (and hence the CVP VoiceXML “value” element data) to be defined. For inline grammars, the CVP VoiceXML Form\_With\_Confirm element automatically generates the grammar slot name based on the slot attribute or the field name. For custom-defined grammars that are referenced from an external source, it is the developer’s job to configure “Field Name” and/or “Field slot” correctly so that the above requirement is met.
- For a grammar that returns either different slots for different inputs or multiple slots per utterance, there are two ways to configure the Form\_With\_Confirm element to store the slot/value pairs:
  - Leave the “slot\_element\_data” setting empty. The Form\_With\_Confirm element will create element data named “nbestInterpretationX” (where X is from 1 to the length of the n-best list) that contains a string that uses delimiters “+” and “.” to separate the multiple slot names from their values. For example: “+Slot1:value1+Slot2:value2...”. A developer would then need to parse this string in a subsequent element to obtain the different slot name and value pairs.
  - Configure the “slot\_element\_data” setting with the names for all the slots that can be returned. The Form\_With\_Confirm element will create a new set of n-best element data to store the recognition results for each slot listed in that setting. The element data will be named as <SLOT\_ELEMENT\_DATA X> (where “SLOT\_ELEMENT\_DATA” is a string identical to the setting value and X is from 1 to the length of the n-best list). For example, if “slot\_element\_data” have two values “city” and “state” and there are three n-best results triggered, then a total of six element data in the names of “city1”, “city2”, “city3”, “state1”, “state2”, and “state3” will be created to store each of the n-best values for the “city” and “state” slots. Note that if n-best processing is disabled by setting **maxnbest** to 1, then only one interpretation will be returned per input and thereby only one element data per slot (“city1” and “state1”) will be created.

## Element Data

Name	Type	Notes
<b>value</b>	string	This stores the value of the VXML field name variable.
<b>value_confidence</b>	float	This stores the confidence score of the captured Form_With_Confirm utterance. When n-best recognition is enabled, this stores the confidence score of the top hypothesis in the n-best list.
<SLOT_ELEMENT_DATA1> <SLOT_ELEMENT_DATA2> ... <SLOT_ELEMENT_DATA <sup>X</sup> >	string	A separate set of element data stores the interpretation values for each filled slot of captured n-best utterances. While the maximum number of <SLOT_ELEMENT_DATA <sup>X</sup> > values is equal to the <b>maxnbest</b> setting value, the actual number of these values available is dependent on speech recognition at runtime, where <SLOT_ELEMENT_DATA1> holds the slot value of the top hypothesis in the n-best list and <SLOT_ELEMENT_DATA <sup>X</sup> > holds the slot value of the last hypothesis. Note that if the <b>slot_element_data</b> setting is blank, these sets of element data will <i>not</i> be created.
<b>nbestLength</b>	int ≥ 1	This stores the number of n-best hypotheses generated by the speech engine.
<b>nbestUtterance1</b> <b>nbestUtterance2</b> ... <b>nbestUtterance<sup>X</sup></b>	string	This set of element data stores the captured n-best utterances. While the maximum number of <b>nbestUtterance<sup>X</sup></b> values is equal to the <b>maxnbest</b> setting value, the actual number of these values available is determined by speech recognition at runtime, where <b>nbestUtterance1</b> holds the utterance of the top hypothesis in the n-best list and <b>nbestUtterance<sup>X</sup></b> holds the utterance of the last hypothesis.
<b>nbestInterpretation1</b> <b>nbestInterpretation2</b> ... <b>nbestInterpretation<sup>X</sup></b>	string	This set of element data stores the interpretations of captured n-best utterances. While the maximum number of <b>nbestInterpretation<sup>X</sup></b> values is equal to the <b>maxnbest</b> setting value, the actual number of these values available is determined by speech recognition at runtime, where <b>nbestInterpretation1</b> holds the interpretation of the top hypothesis in the n-best list and <b>nbestInterpretation<sup>X</sup></b> holds the interpretation of the last hypothesis.
<b>nbestConfidence1</b> <b>nbestConfidence2</b> ... <b>nbestConfidence<sup>X</sup></b>	float	This set of element data stores the confidence scores of captured n-best utterances. While the maximum number of <b>nbestConfidence<sup>X</sup></b> values is equal to the <b>maxnbest</b> setting value, the actual number of these values available is determined by speech recognition at runtime, where <b>nbestConfidence1</b> holds the confidence score of the top hypothesis in the n-best list and <b>nbestConfidence<sup>X</sup></b> holds the confidence score of the last hypothesis.
<b>nbestInputmode1</b> <b>nbestInputmode2</b> ... <b>nbestInputmode<sup>X</sup></b>	string	This set of element data stores the input modes of captured n-best utterances.

\* “SLOT\_ELEMENT\_DATA” is a string identical to the configuration value of the “slot\_element\_data” setting, and X is from 1 to the length of the n-best list. If more than one such value is configured, then multiple sets of element data using the same naming convention will be created.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>max_disconfirmed</b>	The maximum number of disconfirm events has occurred. If the disconfirm max count is 0, this exit state will never occur.
<b>done</b>	The caller input matched the grammar correctly.

## Audio Groups

### Form Data Capture

Name (Label)	Req'd	Max 1	Notes
<b>form_initial_audio_group</b> (Form Initial)	Yes	Yes	Played when the voice element first begins.
<b>form_nomatch_audio_group</b> (Form NoMatch)	No	No	Played when a nomatch event occurs during form input capture. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>form_noinput_audio_group</b> (Form NoInput)	No	No	Played when a noinput event occurs during form input capture. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>form_help_audio_group</b> (Form Help)	No	No	Played when a help event is triggered during form input capture. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### Form Data Confirm

Name (Label)	Req'd	Max 1	Notes
<b>confirm_initial_audio_group</b> (Confirm Initial)	Yes	Yes	Played after the caller enters a value, requesting the caller's confirmation of that value.
<b>confirm_nomatch_audio_group</b> (Confirm NoMatch)	No	No	Played when a nomatch event occurs during confirmation. The nomatch event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a nomatch.
<b>confirm_noinput_audio_group</b> (Confirm NoInput)	No	No	Played when a noinput event occurs during confirmation. The noinput event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a noinput.

Name (Label)	Req'd	Max 1	Notes
<b>confirm_help_audio_group</b> (Confirm Help)	No	No	Played when a help event is triggered during confirmation. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the confirm nomatch audio group to play (or the confirm initial audio group if confirm nomatch audio group is not specified).
<b>disconfirmed_audio_group</b> (Disconfirmed)	No	No	Played when the caller disconfirms the value.

*End*

Name (Label)	Req'd	Max 1	Notes
<b>yes_audio_group</b> (Yes)	No	Yes	Played after the caller chooses the "yes" option. If not specified, no audio will be played when this option is chosen.

**Studio Element Folder:** Form**Class Name:** `com.audium.server.voiceElement.form.MFoundationFormWithConfirm`

## 2\_Option\_Menu, 3\_Option\_Menu, . . . , 10\_Option\_Menu

These voice elements define menus that support from 2 to 10 options. The Menu voice elements are similar to the Form voice element, except that the number of choices is fixed and all grammars are defined within the voice elements. Additionally, there is an exit state for each option, therefore the captured value does not have to be analyzed afterwards to determine the next dialog in the call flow. Use Menu elements when the situation defines a fixed number of choices where each choice does something different in the call flow.

Because the number of exit states is fixed for a voice element, there are separate voice elements for Menu voice elements with 2 to 10 options. For each additional option, three additional settings are added to handle the spoken keyword, DTMF entry, and interpretation value for each option. The audio groups and element data saved are the same for all Menu voice elements.

Each option must be assigned an interpretation value that the element will return as element data named `value` when any of the keywords or DTMF key presses assigned to that option are captured. The element variable (`value`) will contain the same value regardless of the input mode (speech or DTMF).

The audio groups are identical to those of the Form voice element. The `done_audio_group` group may be used for a message that is to be played regardless of what option is chosen. If an `option` specific message is desired, it is recommended that the `done_audio_group` not be configured and an Audio voice element be used after the particular choice is made.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<code>max_noinput_count</code> (Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during input capture. 0 = infinite noinputs allowed.
<code>max_nomatch_count</code> (Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during input capture. 0 = infinite nomatches allowed.
<code>confidence_level</code> (Confidence Level)	decimal (0.0 to 1.0)	Yes	true	true	0.40	The confidence level threshold to use.

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>optionX_dtmf</b> <sup>†</sup> (Option X DTMF)	Character (0-9, #, *) <sup>†</sup>	No	true	true	None	The dtmf keypress or keypresses that can be used to select the menu <b>optionX</b> . Additional <b>optionX_dtmf</b> settings may be used to define multiple dtmf keypresses corresponding to the same return value. Note that 1) at minimum, one of the two settings: <b>optionX_dtmf</b> or <b>optionX_voice</b> <i>must</i> be specified; and 2) keypresses are currently limited to single digits.
<b>optionX_voice</b> (Option X Voice)	string	No	true	true	None	The voice keyword or keywords that can be used to select the menu <b>optionX</b> . Additional <b>optionX_voice</b> settings may be used to define multiple matching voice keywords corresponding to the same return value. Note that at the minimum, one of the two settings: <b>optionX_dtmf</b> or <b>optionX_voice</b> <i>must</i> be specified.
<b>optionX_value</b> (Option X Value)	string	Yes	false	true	None	The value to be stored in the element data <b>value</b> for this voice element when the caller selects <b>optionX</b> . Note that only a single value is allowed for each option.

<sup>†</sup>where X is 2 - 10 as applicable.

## Element Data

Name	Type	Notes
<b>value</b>	string	The value associated with the keyword or DTMF keypress inputted by the caller is stored in this variable.
<b>confidence</b>	float	This is the confidence value of the matched utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the <b>max_nomatch_count</b> is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the <b>max_noinput_count</b> is 0, this exit state will never occur.
<b>optionX</b>	The utterance or DTMF entry matched <b>optionX</b> .

Notes:

- Each option can react on just a spoken keyword, just DTMF keypresses, or both, but at least one method must be specified or an error will be reported.
- All options in the menu must have a consistent input mode. For example, a menu cannot be configured so that option 1 is chosen through both voice and DTMF but option 2 is chosen only through voice.
- There are no menus with more than 10 options. In cases where more are needed, a Form voice element is recommended.

## Audio Groups

### Menu Option Capture

Name (Label)	Req'd	Max 1	Notes
<b>initial_audio_group</b> (Initial)	Yes	Yes	Played when the voice element first begins.
<b>nomatch_audio_group</b> (NoMatch)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>noinput_audio_group</b> (NoInput)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>help_audio_group</b> (Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### End

Name (Label)	Req'd	Max 1	Notes
<b>done_audio_group</b> (Done)	No	Yes	Played when the voice element completes any of the option exit states.

**Studio Element Folder:** Menu

**Class Name:** `com.audium.server.voiceElement.menu.MFoundationXOptionsMenu` (where X is from 2 to 10).

# Number

The Number voice element captures a number input from the caller. The number can be spoken or entered using the keypad. The resulting value will be stored in element data as a decimal value. The number can be negative or positive and can contain a decimal point. Using DTMF entry, however, the number is restricted to being positive and the decimal point is entered by pressing the \* key. Using speech input, the number may be spoken naturally.

## Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	String enum	Yes	true	false	both	The type of entry allowed for input. Possible values are: <code>voice</code>   <code>dtmf</code>   <code>both</code> .
<b>noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<b>max_noinput_count</b> (Number Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during number input capture. 0 = infinite noinputs allowed.
<b>max_nomatch_count</b> (Number Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during number input capture. 0 = infinite nomatches allowed.
<b>number_confidence_level</b> (Number Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use during number capture.

## Element Data

Name	Type	Notes
<b>value</b>	string	The number captured and stored as a whole or decimal number with an optional minus sign.
<b>value_confidence</b>	float	This is the confidence value of the captured utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>done</b>	The number capture was completed.

Notes:

- If the number to be captured is a positive whole number and the input is via DTMF, the number can be entered using this voice element or the **Digits** voice element.

## Audio Groups

### Number Capture

Name (Label)	Req'd	Max 1	Notes
<b>number_initial_audio_group</b> (Number Initial)	Yes	Yes	Played when the voice element first begins.
<b>number_nomatch_audio_group</b> (Number NoMatch)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>number_noinput_audio_group</b> (Number NoInput)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>number_help_audio_group</b> (Number Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### End

Name (Label)	Req'd	Max 1	Notes
<b>done_audio_group</b> (Done)	No	Yes	Played when the number capture is completed and the voice element exits with the <b>done</b> exit state.

**Studio Element Folder:** Number Capture

**Class Name:** `com.audium.server.voiceElement.number.MBasicNumber`

## Number\_With\_Confirm

The Number\_With\_Confirm voice element captures a standard number, and presents a confirmation menu allowing the caller to either accept their entry or re-enter the number. The number can be spoken or entered using the keypad. The resulting value will be stored in element data as a decimal value. The number can be negative or positive and can contain a decimal point. Using DTMF entry, however, the number is restricted to being positive and the decimal point is entered by pressing the \* key. Using speech input, the number may be spoken naturally.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<code>inputmode</code> (Input Mode)	string enum	Yes	true	false	both	The type of entry allowed for input. Possible values are: <code>voice</code>   <code>dtmf</code>   <code>both</code> .
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<code>number_max_noinput_count</code> (Number Max NoInput Count)	int $\geq 0$	Yes	true	true	3	The maximum number of noinput events allowed during number input capture. 0 = infinite noinputs allowed.
<code>number_max_nomatch_count</code> (Number Max NoMatch Count)	int $\geq 0$	Yes	true	true	3	The maximum number of nomatch events allowed during number input capture. 0 = infinite nomatches allowed.
<code>confirm_max_noinput_count</code> (Confirm Max NoInput Count)	int $\geq 0$	Yes	true	true	3	The maximum number of noinput events allowed during number input confirmation. 0 = infinite noinputs allowed.
<code>confirm_max_nomatch_count</code> (Confirm Max NoMatch Count)	int $\geq 0$	Yes	true	true	3	The maximum number of nomatch events allowed during number input confirmation. 0 = infinite nomatches allowed.
<code>max_disconfirmed_count</code> (Max Disconfirmed Count)	int $\geq 0$	Yes	true	true	3	The maximum number of times a caller is allowed to disconfirm a captured input. 0 = infinite disconfirmations allowed.
<code>number_confidence_level</code> (Number Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use during number capture.
<code>confirm_confidence_level</code> (Confirm Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.50	The confidence level threshold to use during confirmation.

## Element Data

Name	Type	Notes
<b>value</b>	string	The number captured and stored as a whole or decimal number with an optional minus sign.
<b>value_confidence</b>	float	This is the confidence value of the captured number utterance.
<b>confirm_confidence</b>	float	This is the confidence value of the captured confirm utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>max_disconfirmed</b>	The maximum number of disconfirmation has occurred. If the max disconfirmed count is set to 0, this exit state will never occur.
<b>done</b>	The number captured was confirmed.

Notes:

- If the number to be captured is a positive whole number and the input is via DTMF, the number can be entered using this voice element or the **Digits\_With\_Confirm** voice element.

## Audio Groups

### Number Capture

Name (Label)	Req'd	Max 1	Notes
<b>number_initial_audio_group</b> (Number Initial)	Yes	Yes	Played when the voice element first begins.
<b>number_nomatch_audio_group</b> (Number NoMatch)	No	No	Played when a nomatch event occurs during number input capture. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>number_noinput_audio_group</b> (Number NoInput)	No	No	Played when a noinput event occurs during number input capture. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>number_help_audio_group</b> (Number Help)	No	No	Played when a help event is triggered during number input capture. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group

Name (Label)	Req'd	Max 1	Notes
			is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

*Number Confirm*

Name (Label)	Req'd	Max 1	Notes
<b>confirm_initial_audio_group</b> (Confirm Initial)	Yes	Yes	Played when confirmation first begins.
<b>confirm_nomatch_audio_group</b> (Confirm NoMatch)	No	No	Played when a nomatch event occurs during confirmation. The nomatch event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a nomatch.
<b>confirm_noinput_audio_group</b> (Confirm NoInput)	No	No	Played when a noinput event occurs during confirmation. The noinput event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a noinput.
<b>confirm_help_audio_group</b> (Confirm Help)	No	No	Played when a help event is triggered during confirmation. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the confirm nomatch audio group to play (or the confirm initial audio group if confirm nomatch audio group is not specified).
<b>disconfirmed_audio_group</b> (Disconfirmed)	No	No	Played after the caller disconfirms a captured number entry.

*End*

Name (Label)	Req'd	Max 1	Notes
<b>yes_audio_group</b> (Yes)	No	Yes	Played after the caller chooses the "yes" option. If not specified, no audio will be played when this option is chosen.

**Studio Element Folder:** Number Capture**Class Name:** com.audium.server.voiceElement.number.MBasicNumberWithConfirm

## Phone

The Phone voice element captures a phone number input from the caller. The phone number can be spoken or entered using the keypad. The captured value will be stored in element data as a string. The string may contain a number of digits and an optional character “x” to indicate a phone number with an extension. Using speech input, the entire phone number (including the extension) may be spoken in natural language. Using DTMF entry, the caller can enter an extension by pressing the \* keypress followed by the extension.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<code>inputmode</code> (Input Mode)	string enum	Yes	true	false	<i>both</i>	The type of entry allowed for input. Possible values are: <code>voice</code>   <code>dtmf</code>   <code>both</code> .
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	<i>5s</i>	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<code>collect_max_noinput_count</code> (Phone Max NoInput Count)	int $\geq 0$	Yes	true	true	3	The maximum number of noinput events allowed during phone input capture. 0 = infinite noinputs allowed.
<code>collect_max_nomatch_count</code> (Phone Max NoMatch Count)	int $\geq 0$	Yes	true	false	3	The maximum number of nomatch events allowed during phone input capture. 0 = infinite nomatches allowed.
<code>collect_confidence_level</code> (Phone Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use during phone capture.

### Element Data

Name	Type	Notes
<code>value</code>	string	The phone number captured.
<code>value_confidence</code>	float	This is the confidence value of the captured phone utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the max nomatch count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the max noinput count is 0, this exit state will never occur.
<b>done</b>	The phone number capture was completed.

## Audio Groups

### Phone Capture

Name (Label)	Req'd	Max 1	Notes
<b>collect_initial_audio_group</b> (Phone Initial)	Yes	Yes	Played when the voice element first begins.
<b>collect_noinput_audio_group</b> (Phone NoInput)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>collect_nomatch_audio_group</b> (Phone NoMatch)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>collect_help_audio_group</b> (Phone Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### End

Name (Label)	Req'd	Max 1	Notes
<b>done_audio_group</b> (Done)	No	Yes	Played after phone capture is completed.

### Studio Element Folder: Number Capture

**Class Name:** `com.audium.server.voiceElement.phone.MBasicPhone`

## Phone\_With\_Confirm

The Phone\_With\_Confirm voice element captures a phone number input from the caller, and presents a confirmation menu allowing the caller to either accept their entry or re-enter the phone number. The phone number can be spoken or entered using the keypad. The captured value will be stored in element data as a string. The string may contain a number of digits and an optional character “x” to indicate a phone number with an extension. Using speech input, the entire phone number (including the extension) may be spoken in natural language. Using DTMF entry, the caller can enter an extension by pressing the \* keypress followed by the extension.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<code>inputmode</code> (Input Mode)	string enum	Yes	true	false	<i>both</i>	The type of entry allowed for input. Possible values are: <code>voice</code>   <code>dtmf</code>   <code>both</code> .
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	<i>5s</i>	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<code>collect_max_noinput_count</code> (Phone Max NoInput Count)	$\text{int} \geq 0$	Yes	true	true	3	The maximum number of noinput events allowed during phone input capture. 0 = infinite noinputs allowed.
<code>collect_max_nomatch_count</code> (Phone Max NoMatch Count)	$\text{int} \geq 0$	Yes	true	false	3	The maximum number of nomatch events allowed during phone input capture. 0 = infinite nomatches allowed.
<code>confirm_max_noinput_count</code> (Confirm Max NoInput Count)	$\text{int} \geq 0$	Yes	true	true	3	The maximum number of noinput events allowed during phone input confirmation. 0 = infinite noinputs allowed.
<code>confirm_max_nomatch_count</code> (Confirm Max NoMatch Count)	$\text{int} \geq 0$	Yes	true	false	3	The maximum number of nomatch events allowed during phone input confirmation. 0 = infinite nomatches allowed.
<code>max_disconfirmed_count</code> (Max Disconfirmed Count)	$\text{int} \geq 0$	Yes	true	false	3	The maximum number of times a caller is allowed to disconfirm a captured input. 0 = infinite disconfirmations allowed.
<code>collect_confidence_level</code> (Phone Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use during phone capture.
<code>confirm_confidence_level</code> (Confirm Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.50	The confidence level threshold to use during confirmation.

## Element Data

Name	Type	Notes
<b>value</b>	string	The phone number captured.
<b>value_confidence</b>	float	This is the confidence value of the captured phone number utterance.
<b>confirm_confidence</b>	float	This is the confidence value of the captured confirm utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events occurred. If the max nomatch count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events occurred. If the max noinput count is 0, this exit state will never occur.
<b>max_disconfirmed</b>	The maximum number of disconfirmation occurred. If the max disconfirmed count is set to 0, this exit state will never occur.
<b>done</b>	The phone number captured was confirmed.

## Audio Groups

### Phone Capture

Name (Label)	Req'd	Max 1	Notes
<b>collect_initial_audio_group</b> (Phone Initial)	Yes	Yes	Played when the voice element first begins.
<b>collect_nomatch_audio_group</b> (Phone NoMatch)	No	No	Played when a nomatch event occurs during phone input capture. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>collect_noinput_audio_group</b> (Phone NoInput)	No	No	Played when a noinput event occurs during phone input capture. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>collect_help_audio_group</b> (Phone Help)	No	No	Played when a help event is triggered during phone input capture. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

*Phone Confirm*

Name (Label)	Req'd	Max 1	Notes
<b>confirm_initial_audio_group</b> (Confirm Initial)	Yes	Yes	Played when confirmation first begins.
<b>confirm_nomatch_audio_group</b> (Confirm NoMatch)	No	No	Played when a nomatch event occurs during confirmation. The nomatch event count corresponds to the audio group count. If not specified, the <b>confirm_initial_audio_group</b> will be played in the event of a nomatch.
<b>confirm_noinput_audio_group</b> (Confirm NoInput)	No	No	Played when a noinput event occurs during confirmation. The noinput event count corresponds to the audio group count. If not specified, the <b>confirm_initial_audio_group</b> will be played in the event of a noinput.
<b>confirm_help_audio_group</b> (Confirm Help)	No	No	Played when a help event is triggered during confirmation. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the <b>confirm_nomatch_audio_group</b> to play (or the <b>confirm_initial_audio_group</b> if <b>confirm_nomatch_audio_group</b> is not specified).
<b>disconfirmed_audio_group</b> (Disconfirmed)	No	No	Played after the caller disconfirms a captured phone entry. Upon reaching the <b>max_disconfirmed_count</b> , the prompt content should be about exiting with the <b>max_disconfirmed</b> exit state.

*End*

Name (Label)	Req'd	Max 1	Notes
<b>yes_audio_group</b> (Yes)	No	Yes	Played after the caller chooses the "yes" option. If not specified, no audio will be played when this option is chosen.

**Studio Element Folder:** Number Capture**Class Name:** com.audium.server.voiceElement.phone.MBasicPhoneWithConfirm

## Record

The Record voice element makes a recording of the caller's voice. A prompt is played to the caller then the voice element records the caller's voice until a termination key is inputted or the recording time limit has been reached. An audio cue (beep) may be activated to signal to the caller that the system is ready to record the caller's voice.

The recording is sent to the Record element by the voice browser and is stored in an audio file in a location specified by the developer. Any pre-existing file with the same name will be overwritten. The element can be configured to produce a non-repeating filename so all recordings can be retained. By default the format for this filename is *audioN.wav* where N is the number of milliseconds since midnight January 1, 1970 (GMT), and all recordings are saved in the WAV format.

## Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<code>max_noinput_count</code> (Max NoInput Count)	int $\geq$ 0	Yes	true	true	3	The maximum number of noinput events allowed during input capture. 0 = infinite noinputs allowed.
<code>start_with_beep</code> (Start With Beep)	boolean	Yes	true	true	true	Whether or not to play a beep before recording begins.
<code>terminate_on_dtmf</code> (Terminate On DTMF)	boolean	Yes	true	true	true	Whether or not the caller can end the recording by pressing a touchtone key.
<code>max_record_time</code> (Max Record Time)	int $\geq$ 0	Yes	true	true	180	The maximum number of seconds the recording is allowed to last.
<code>final_silence</code> (Final Silence)	string	Yes	true	true	4s	The interval of silence (in number of seconds) that indicates the end of speech.
<code>filename</code> (Filename)	string	No	true	true	None	The filename of the recording (without extension). If left blank, an auto-generated filename will be used.
<code>file_type</code> (File Type)	string enum	Yes	true	true	wav	This specifies the audio type of the file that will hold the recording. Possible values are: wav   vox   au   other.

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>mime_type</b> (Mime Type)	string	Yes	true	true	<i>none</i>	This specifies the MIME type of the file that will hold the recording, if <b>file_type</b> is set to other.
<b>file_extension</b> (File Extension)	string	No	true	true	<i>None</i>	This specifies the file extension to use for the recorded file. A file extension different from the file type can be used. For example, with a mime type of vox, the file extension could be set to "ulaw".
<b>path</b> (Path)	string	No	true	true	<i>None</i>	The path to the file that will hold the recording. If left blank, the file is assumed to be sent via ftp.
<b>ftp_host</b> (FTP Host)	string	No	true	true	<i>None</i>	The domain name of the host to ftp the recording. If left blank, the recording is assumed to be stored in a local file.
<b>ftp_user</b> (FTP User)	string	No	true	true	<i>None</i>	The user name to use while FTPing the recording.
<b>ftp_password</b> (FTP Password)	string	No	true	true	<i>None</i>	The password to use while FTPing the recording.
<b>ftp_path</b> (FTP Path)	string	No	true	true	<i>None</i>	The directory in which to FTP the recording.
<b>ftp_in_background</b> (FTP In Background)	boolean	Yes	true	true	true	Whether or not the FTP is to be performed in the background.

## Notes:

- Nomatch events cannot be thrown in this voice element. Since all audio is recorded (except DTMF key presses), there is no reaction on spoken commands (including hotlinks).
- A noinput event is possible if the voice browser detects no audio once recording has started. If the input timeout has been reached, the noinput event is thrown.
- The path setting does not require a trailing slash. The voice element will determine the appropriate destination. The path may be specified in operating system specific format (for example, on Windows it might be specified as *C:\directory\subdirectory\*).
- If **terminate\_on\_DTMF** is “false” or off, recording will stop only after the voice browser reaches the input timeout.
- It is important to ensure that CVP VoiceXML Server has permission to save audio files to the specified path.

## Element Data

Name	Type	Notes
<b>filename</b>	string	This stores the filename of the recording (without the path).
<b>filepath</b>	string	This stores the path to the file holding the recording (including the filename).

## Exit States

Name	Notes
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the <b>max_noinput_count</b> is set to 0, this exit state will never occur.
<b>done</b>	The message was recorded.

## Audio Groups

### *Record Capture*

Name (Label)	Req'd	Max 1	Notes
<b>initial_audio_group</b> (Initial)	Yes	Yes	Played when the voice element first begins.
<b>noinput_audio_group</b> (NoInput)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.

### Studio Element Folder: Record

**Class Name:** `com.audium.server.voiceElement.record.MRecord`

## Record\_With\_Confirm

The Record\_With\_Confirm voice element combines the functionality of the Record voice element with that of the Yes\_No\_Menu voice element. The voice element records the caller's voice, and prompts the caller to confirm that the recording is acceptable. The caller can then accept or reject the confirmation or ask to have the message replayed. If the caller accepts the recording, the voice element saves the file the same way as the Record voice element. This voice element contains all the settings and audio groups from the Record and Yes\_No\_Menu voice elements, however audio groups that are found in the latter two voice elements (nomatch, noinput, and help) are now renamed for them to be distinguished.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<code>inputmode</code> (Input Mode)	string enum	Yes	true	true	both	The type of entry allowed for input during confirmation. Possible values are: voice   dtmf   both.
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<code>record_max_noinput_count</code> (Record Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during input capture. 0 = infinite noinputs allowed.
<code>confirm_max_noinput_count</code> (Confirm Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during confirmation. 0 = infinite noinputs allowed.
<code>confirm_max_nomatch_count</code> (Confirm Max NoMatch Count)	int ≥ 0	Yes	true	true	3	The maximum number of nomatch events allowed during confirmation. 0 = infinite nomatches allowed.
<code>max_disconfirmed_count</code> (Max Disconfirmed Count)	int ≥ 0	Yes	true	true	3	The maximum number of times a caller is allowed to reject a recording. 0 = infinite disconfirmations allowed.
<code>confirm_confidence_level</code> (Confirm Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.50	The confidence level threshold to use for the confirmation.
<code>start_with_beep</code> (Start With Beep)	boolean	Yes	true	true	true	Whether or not to play a beep before recording begins.

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>terminate_on_dtmf</b> (Terminate On DTMF)	boolean	Yes	true	true	true	Whether or not the caller can end the recording by pressing a touchtone key.
<b>max_record_time</b> (Max Record Time)	int ≥ 0	Yes	true	true	180	The maximum number of seconds the recording is allowed to last.
<b>final_silence</b> (Final Silence)	string	Yes	true	true	4s	The interval of silence (in number of seconds) that indicates the end of speech.
<b>replay</b> (Replay)	boolean	Yes	true	true	false	Adds an option to replay the confirm initial audio groups.
<b>filename</b> (Filename)	string	No	true	true	None	The filename of the recording (without extension). If left blank, an auto-generated filename will be used.
<b>file_type</b> (File Type)	string enum	Yes	true	true	wav	This specifies the audio type of the file that will hold the recording. Possible values are: wav   vox   au   other.
<b>mime_type</b> (Mime Type)	string	Yes	true	true	none	This specifies the MIME type of the file that will hold the recording, if <b>file_type</b> is set to other.
<b>file_extension</b> (File Extension)	string	No	true	true	None	This specifies the file extension to use for the recorded file. A file extension different from the file type can be used. For example, with a mime type of vox, the file extension could be set to "ulaw".
<b>path</b> (Path)	string	No	true	true	None	The path to the file that will hold the recording. If left blank, the file is assumed to be sent via ftp.
<b>ftp_host</b> (FTP Host)	string	No	true	true	None	The domain name of the host to ftp the recording. If left blank, the recording is assumed to be stored in a local file.
<b>ftp_user</b> (FTP User)	string	No	true	true	None	The user name to use while FTPing the recording.
<b>ftp_password</b> (FTP Password)	string	No	true	true	None	The password to use while FTPing the recording.
<b>ftp_path</b> (FTP Path)	string	No	true	true	None	The directory in which to FTP the recording.
<b>ftp_in_background</b> (FTP In Background)	boolean	Yes	true	true	true	Whether or not the FTP is to be performed in the background.

## Notes:

- The path setting does not require a trailing slash. The voice element will determine the appropriate destination. The path may be specified in operating system specific format (for example, on Windows it might be specified as *C:\directory\subdirectory\*).
- If `terminate_on_DTMF` is “false” or off, recording will stop only after the voice browser reaches the input timeout.
- It is important to ensure that CVP VoiceXML Server has permission to save audio files to the specified path.

## Element Data

Name	Type	Notes
<code>filename</code>	string	This stores the filename of the recording (without the path).
<code>filepath</code>	string	This stores the path to the file holding the recording (including the filename).
<code>confirm_confidence</code>	float	This is the confidence value of the utterance for the confirmation menu.

## Exit States

Name	Notes
<code>max_nomatch</code>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<code>max_noinput</code>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<code>max_disconfirmed</code>	The maximum number of disconfirms has occurred. If the maximum disconfirm count is 0, this exit state will never occur.
<code>done</code>	The recorded message was confirmed.

## Audio Groups

*Record Capture*

Name (Label)	Req'd	Max 1	Notes
<code>record_initial_audio_group</code> (Record Initial)	Yes	Yes	Played when the voice element first begins.
<code>record_noinput_audio_group</code> (Record NoInput)	No	No	Played when a noinput event occurs during recording. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.

*Record Confirm*

Name (Label)	Req'd	Max 1	Notes
<b>before_confirm_audio_group</b> (Before Confirm)	No	Yes	Played before the recording is played back. The recording will be played back after this audio group is done playing.
<b>after_confirm_audio_group</b> (After Confirm)	No	Yes	Played after the recording is played back. At least one of the two confirm prompts must be specified.
<b>confirm_nomatch_audio_group</b> (Confirm NoMatch)	No	No	Played when a nomatch event occurs during confirmation. The nomatch event count corresponds to the audio group count. If not specified, the confirm initial audio groups will be played in the event of a nomatch.
<b>confirm_noinput_audio_group</b> (Confirm NoInput)	No	No	Played when a noinput event occurs during confirmation. The noinput event count corresponds to the audio group count. If not specified, the confirm initial audio groups will be played in the event of a noinput.
<b>confirm_help_audio_group</b> (Confirm Help)	No	No	Played when a help event is triggered during confirmation. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the confirm nomatch audio group to play (or the confirm initial audio group if confirm nomatch audio group is not specified).
<b>max_disconfirmed_audio_group</b> (Max Disconfirmed)	No	Yes	Played after the caller disconfirms the recorded entry, upon reaching the <b>max_disconfirmed_count</b> . The prompt should be about exiting with the <b>max_disconfirmed</b> exit state.

**Studio Element Folder:** Record**Class Name:** `com.audium.server.voiceElement.record.MRecordWithConfirm`

## Subdialog\_Return

The Subdialog\_Return element is used to return data back to the calling subdialog. This element must be used in all parts of the voice application when the application is complete or the call is directed to leave the application unless the application is designed to be used via a Cisco CVP VoiceXML Application to Application transfer element. Because all voice elements must have at least a single exit state, the “done” exit state for this element should always be mapped to a hangup element. Only the Subdialog\_Return element should be configured to have an exit state that goes to a hangup. The settings for this element are used to define what data to pass back to the calling subdialog. The settings will populate ICM ECC variables associated with data returned to the subdialog. The data must conform to the requirements of the ICM ECC variables, refer to the Cisco CVP documentation for further detail. Each element setting corresponds to the ICM ECC external variable name. The settings should be configured with the data that is to be passed back to ICM. Note, the “caller\_input” setting must be assigned a value in order for the application to validate as CVP requires this variable to have a value.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>caller_input</b> ( <b>Caller Input</b> )	string	yes	true	true	<i>none</i>	Cisco required return argument. Developer must specify a value that is returned to the calling dialog/CVP.
<b>FromExtVXML0</b> ( <b>External VXML 0</b> )	string	no	true	true	<i>none</i>	Optional return argument that is returned to the calling dialog/CVP.
<b>FromExtVXML1</b> ( <b>External VXML 1</b> )	string	no	true	true	<i>none</i>	Optional return argument that is returned to the calling dialog/CVP.
<b>FromExtVXML2</b> ( <b>External VXML 2</b> )	string	no	true	true	<i>none</i>	Optional return argument that is returned to the calling dialog/CVP.
<b>FromExtVXML3</b> ( <b>External VXML 3</b> )	string	no	true	true	<i>none</i>	Optional return argument that is returned to the calling dialog/CVP.

### Exit States

Name	Notes
<b>done</b>	The element execution is complete

**Studio Element Folder:** Cisco

**Class Name:** `com.audium.server.customelement.cisco.CiscoSubdialogReturnElement`

## Subdialog\_Start

The Subdialog\_Start element should be placed at the entrance point of any application after the Call Start element, unless the application is designed to be invoked by a Cisco CVP VoiceXML Application to Application transfer element. This element is used to take data passed from the calling subdialog and create corresponding element or session data. Data can be passed from the calling subdialog to the application as HTTP parameters and/or as VoiceXML parameters (using the <param> tag). If data from the calling subdialog is passed as HTTP parameters, CVP VoiceXML Server will automatically create session data using the name of the data that is sent. However, if data is passed using VoiceXML parameters, the Subdialog\_Start element must be configured appropriately in order for the data to be available as element or session data for the duration of the call session. For each parameter that is passed to the application from the calling subdialog, the “Parameter” setting should be configured with the name of the data that is passed. The “Store As” setting allows developers to choose to store all passed values as session or element data. The “Enable Digits Bypass” setting is used to activate a VoiceXML workaround to ensure expected functionality for a particular TDM or analog phone and flase for IP phones.. If this setting is activated, set to “true”, the “Audio Filler URI” setting can be configured to set a reference to the Flash: silence\_.1s.wav file.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>Parameter (Parameter)</b>	string	no	false	true	<i>none</i>	Holds the name of a parameter passed as input to the subdialog. It must match the exact value specified in the VoiceXML page that calls the subdialog. This is a repeatable setting, so multiple values can be specified.
<b>Where (Store As)</b>	string	no	true	false	Session Data	Determines whether the parameter passed to the subdialog will be stored as element data or session data. By making it element data, the information will “belong” only to this element, and so there is no chance that these variables will overwrite any other variables.
<b>enable_digits_bypass (Enable Digits Bypass)</b>	boolean	yes	true	true	false	Determines whether the digits field is used at the beginning of an application. By default this is disabled.
<b>audio_filler_uri (Audio Filler URI)</b>	string	no	true	true	<i>none</i>	This setting allows to configure an audio file for the prompt in the digits field.

---

## Exit States

Name	Notes
done	The element execution is complete

**Studio Element Folder:** Cisco

**Class Name:** com.audium.server.customelement.cisco.CiscoSubdialogEntranceElement

## Time

The Time voice element captures a time input from the caller. The time input can be entered using spoken inputs (including hours and minutes) or DTMF inputs (in the HHMM format). The captured value will be stored in element data as a five character string in the format HHMMX, where X is one of four possible values: “a” for AM, “p” for PM, “h” for a military time, or “?” for an ambiguous time. Using speech input, the time input may be spoken in natural language.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	string enum	Yes	true	false	<i>both</i>	The type of entry allowed for input. Possible values are: <i>voice</i>   <i>dtmf</i>   <i>both</i> .
<b>noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	<i>5s</i>	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<b>collect_max_noinput_count</b> (Time Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during time input capture. 0 = infinite noinputs allowed.
<b>collect_max_nomatch_count</b> (Time Max NoMatch Count)	int ≥ 0	Yes	true	false	3	The maximum number of nomatch events allowed during time input capture. 0 = infinite nomatches allowed.
<b>collect_confidence_level</b> (Time Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use during time capture.

### Element Data

Name	Type	Notes
<b>value</b>	string	The time stored in the HHMMX format, where X is one of a, p, h, or ?.
<b>value_confidence</b>	float	This is the confidence value of the captured time utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>done</b>	The time capture was completed.

## Audio Groups

### Time Capture

Name (Label)	Req'd	Max 1	Notes
<b>collect_initial_audio_group</b> (Time Initial)	Yes	Yes	Played when the voice element first begins.
<b>collect_noinput_audio_group</b> (Time NoInput)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>collect_nomatch_audio_group</b> (Time NoMatch)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>collect_help_audio_group</b> (Time Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

### End

Name (Label)	Req'd	Max 1	Notes
<b>done_audio_group</b> (Done)	No	Yes	Played after the time capture is completed. If not specified, no audio will be played.

### Studio Element Folder: Date & Time

**Class Name:** com.audium.server.voiceElement.time.MBasicTime

## Time\_With\_Confirm

The Time\_With\_Confirm voice element captures a time input from the caller, and presents a confirmation menu allowing the caller to either accept their entry or re-enter the time. The time input can be entered using spoken inputs (including hours and minutes) or DTMF inputs (in the HHMM format). The captured value will be stored in element data as a five character string in the format HHMMX, where X is one of four possible values: “a” for AM, “p” for PM, “h” for a military time, or “?” for an ambiguous time. Using speech input, the time input may be spoken in natural language.

### Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>inputmode</b> (Input Mode)	string enum	Yes	true	false	<i>both</i>	The type of entry allowed for input. Possible values are: <i>voice</i>   <i>dtmf</i>   <i>both</i> .
<b>noinput_timeout</b> (Noinput Timeout)	string	Yes	true	true	<i>5s</i>	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<b>collect_max_noinput_count</b> (Time Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during time input capture. 0 = infinite noinputs allowed.
<b>collect_max_nomatch_count</b> (Time Max NoMatch Count)	int ≥ 0	Yes	true	false	3	The maximum number of nomatch events allowed during time input capture. 0 = infinite nomatches allowed.
<b>confirm_max_noinput_count</b> (Confirm Max NoInput Count)	int ≥ 0	Yes	true	true	3	The maximum number of noinput events allowed during time input confirmation. 0 = infinite noinputs allowed.
<b>confirm_max_nomatch_count</b> (Confirm Max NoMatch Count)	int ≥ 0	Yes	true	false	3	The maximum number of nomatch events allowed during time input confirmation. 0 = infinite nomatches allowed.
<b>max_disconfirmed_count</b> (Max Disconfirmed Count)	int ≥ 0	Yes	true	false	3	The maximum number of times a caller is allowed to disconfirm a captured input. 0 = infinite disconfirmations allowed.
<b>collect_confidence_level</b> (Time Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.40	The confidence level threshold to use during time capture.
<b>confirm_confidence_level</b> (Confirm Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.50	The confidence level threshold to use during confirmation.

## Element Data

Name	Type	Notes
<b>value</b>	string	The time stored in the HHMMX format, where X is one of a, p, h, or ?.
<b>value_confidence</b>	float	This is the confidence value of the captured time utterance.
<b>confirm_confidence</b>	float	This is the confidence value of the captured confirm utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>max_disconfirmed</b>	The maximum number of disconfirmation has occurred. If the <b>max_disconfirmed_count</b> is set to 0, this exit state will never occur.
<b>done</b>	The time captured is confirmed.

## Audio Groups

### Time Capture

Name (Label)	Req'd	Max 1	Notes
<b>collect_initial_audio_group</b> (Time Initial)	Yes	Yes	Played when the voice element first begins.
<b>collect_noinput_audio_group</b> (Time NoInput)	No	No	Played when a nomatch event occurs during time input capture. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>collect_nomatch_audio_group</b> (Time NoMatch)	No	No	Played when a noinput event occurs during time input capture. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>collect_help_audio_group</b> (Time Help)	No	No	Played when a help event is triggered during time input capture. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch

Name (Label)	Req'd	Max 1	Notes
			audio group is not specified).

*Time Confirm*

Name (Label)	Req'd	Max 1	Notes
<b>confirm_initial_audio_group</b> (Confirm Initial)	Yes	Yes	Played when confirmation of the captured time first begins.
<b>confirm_nomatch_audio_group</b> (Confirm NoMatch)	No	No	Played when a nomatch event occurs during confirmation. The nomatch event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a nomatch.
<b>confirm_noinput_audio_group</b> (Confirm NoInput)	No	No	Played when a noinput event occurs during confirmation. The noinput event count corresponds to the audio group count. If not specified, the confirm initial audio group will be played in the event of a noinput.
<b>confirm_help_audio_group</b> (Confirm Help)	No	No	Played when a help event is triggered during confirmation. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the confirm nomatch audio group to play (or the confirm initial audio group if confirm nomatch audio group is not specified).
<b>disconfirmed_audio_group</b> (Disconfirmed)	No	No	Played after the caller disconfirms a time entry captured.

*End*

Name (Label)	Req'd	Max 1	Notes
<b>yes_audio_group</b> (Yes)	No	Yes	Played after the caller chooses the "yes" option. If not specified, no audio will be played when this option is chosen.

**Studio Element Folder:** Date & Time**Class Name:** com.audium.server.voiceElement.time.MBasicTimeWithConfirm

## Transfer

The Transfer voice element performs a call transfer to a phone number specified by a configuration setting. Depending on how the voice browser is configured, the call transfer can be a bridge transfer or a blind transfer. For a bridge transfer, the voice browser makes an outbound call while maintaining the original call and acts as a bridge between the two calls. The advantage of this is that once the secondary call ends, the original call can still continue with the IVR. The disadvantage is that two separate phone lines are used. For a blind transfer, the voice browser makes an outbound call to the callee and leaves the connection regardless of the outcome. At this point, the voice browser (and as a result CVP VoiceXML Server) is no longer in control of the call. Blind transfers involve only one line.

The Transfer element defines exit states for the different ways bridge transfers can end such as the person being called hung up, there was no answer, there was a busy signal, or some other phone-related error occurred. Since blind transfers take the call away from the voice browser and CVP VoiceXML Server, a Transfer element performing a blind transfer would never return an exit state. Instead, a special event would be thrown by the voice browser, caught in the root document for the call, and CVP VoiceXML Server would terminate the session by interrupting the Transfer element.

The number to transfer to can be any phone number allowed by the voice browser telephony provider. Check the browser documentation for specific requirements and restrictions for call transfer.

## Settings

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>transfer_destination</b> (Transfer Destination)	string	Yes	true	true	<i>None</i>	The phone number to transfer to. It may contain non-numerical characters to allow support for phone extensions.
<b>destination_type</b> (Destination Type)	string	Yes	true	True	<i>tel</i>	The type of transfer destination to which the voice element is to connect. Possible values are: <i>tel</i>   <i>sip</i> .
<b>connect_timeout</b> (Connect Timeout)	int > 0	Yes	true	true	60	The number of seconds the voice element is allowed to wait for an answer, after which it will exit with the "noanswer" exit state. Note that this setting only applies to the bridge mode.
<b>max_transfer_time</b> (Max Transfer Time)	int > 0	No	true	true	<i>Browser default</i>	The maximum duration (in seconds) the transfer is allowed to last. By default, it is an infinite amount of time, so the setting should be configured only if a limit is required. Note that <i>max_transfer_time</i> only applies to the bridge mode.

Name (Label)	Type	Req'd	Single setting value	Substitution allowed	Default	Notes
<b>bridge</b> ( <b>Bridge</b> )	binary	Yes	true	true	true	This setting determines what to do after the call is connected. Possible values: <code>true</code>   <code>false</code> . When set to <code>true</code> , the caller resumes the interpreter session after the call with the third party completes (a bridge transfer). When set to <code>false</code> , a <code>telephone.disconnect.transfer</code> event is thrown and the interpreter session is terminated, as soon as the call with the third party connects (a blind transfer).
<b>transfer_audio</b> ( <b>Transfer Audio</b> )	string	No	true	true	<i>None</i>	The URI location of the audio file to be played while connecting the call.
<b>aai</b> ( <b>Application-to-application Information</b> )	string	No	true	true	<i>None</i>	A string containing Application-to-Application Information data to be sent to an application on the far-end.

## Element Data

Name	Type	Notes
<b>result</b>	string	The value returned by the transfer field. This is dependent on the voice browser.

## Exit States

Name	Notes
<b>busy</b>	The number was busy.
<b>noanswer</b>	There was no answer.
<b>phone_error</b>	There was some sort of phone-related error.
<b>done</b>	The call transfer completed successfully.

## Notes:

- Hosting voice browsers may disable call transfers for developer accounts. You should verify with your provider that transfer is enabled for your application.
- Some voice browsers use a code to indicate which call transfers will be allowed. This code appears before the phone number.
- Some voice browsers support the inclusion of an extension in the phone number so that the system can transfer to a particular extension. It is up to the developer to pass this voice element a string containing the appropriate format. Check the platform specific documentation for support of extension dialing in transfer.

**Audio Groups***Transfer Audio*

Name (Label)	Req'd	Max 1	Notes
<b>initial_audio_group</b> (Initial)	No	Yes	Played to introduce the transfer. If there is none, the transfer occurs immediately.
<b>busy_audio_group</b> (Busy)	No	Yes	Played when there is a busy signal, right before the voice element exits with the "busy" exit state.
<b>noanswer_audio_group</b> (No Answer)	No	Yes	Played when there is no answer, right before the voice element exits with the "noanswer" exit state.
<b>phone_error_audio_group</b> (Phone Error)	No	Yes	Played when there is some kind of phone-related error, right before the voice element exits with the "phone error" exit state.

*End*

Name (Label)	Req'd	Max 1	Notes
<b>done_audio_group</b> (Done)	No	Yes	Played when the call transfer completes with the party called hanging up and the caller staying on the line.

**Studio Element Folder:** Call Control**Class Name:** `com.audium.server.voiceElement.transfer.MTransfer`

## Yes\_No\_Menu

The Yes\_No\_Menu voice element presents a yes/no menu. It can be configured to accept DTMF entry (1 for yes and 2 for no) or spoken input ("yes" or "no" and other synonymous utterances, however this is dependent on the voice browser). There is an optional feature that allows the word "replay" to be spoken (or DTMF button 3) that replays the `initial_audio_group`. The voice element uses the browser specific VoiceXML builtin grammar for the boolean field type. A separate exit state exists for the yes and no choices (there is no exit state for replay since dialog execution is still contained within the confines of the voice element).

### Settings

Name (Label)	Type	Req'd	Single setting value	Sugstitution allowed	Default	Notes
<code>inputmode</code> (Input Mode)	string enum	Yes	true	false	both	The type of entry allowed for input (using speech recognition, DTMF entry, or both). Possible values are: <code>voice   dtmf   both</code> .
<code>noinput_timeout</code> (Noinput Timeout)	string	Yes	true	true	5s	The maximum time length allowed (in seconds) for silence or no keypress before a noinput event is thrown.
<code>max_noinput_count</code> (Max NoInput Count)	int ≥ 0	Yes	true	true	3	0 = infinite noinputs allowed.
<code>max_nomatch_count</code> (Max NoMatch Count)	int ≥ 0	Yes	true	true	3	0 = infinite nomatches allowed.
<code>confidence_level</code> (Confidence Level)	decimal (0.0 – 1.0)	Yes	true	true	0.50	The confidence level threshold to use.
<code>replay</code> (Replay)	boolean	Yes	true	true	false	True adds a "replay" option which replays the initial prompt.

### Element Data

Name	Type	Notes
<code>value</code>	string	This is the value chosen by the caller. Can be: "yes" or "no".
<code>confidence</code>	float	This is the confidence value of the utterance.

## Exit States

Name	Notes
<b>max_nomatch</b>	The maximum number of nomatch events has occurred. If the nomatch max count is 0, this exit state will never occur.
<b>max_noinput</b>	The maximum number of noinput events has occurred. If the noinput max count is 0, this exit state will never occur.
<b>yes</b>	The utterance was recognized as "yes".
<b>no</b>	The utterance was recognized as "no".

Notes:

- The replay option, when activated, resets all the event counts (noinput and nomatch).

## Audio Groups

*Yes / No Capture*

Name (Label)	Req'd	Max 1	Notes
<b>initial_audio_group</b> (Initial)	Yes	Yes	Played when the voice element first begins.
<b>nomatch_audio_group</b> (NoMatch)	No	No	Played when a nomatch event occurs. The nomatch event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a nomatch.
<b>noinput_audio_group</b> (NoInput)	No	No	Played when a noinput event occurs. The noinput event count corresponds to the audio group count. If not specified, the initial audio group will be played in the event of a noinput.
<b>help_audio_group</b> (Help)	No	No	Played when a help event is triggered. The help event count corresponds to the audio group count. By default, the help event can be triggered by the voice input "help". Additionally, other voice and DTMF inputs can be used to trigger the help event by configuring a hotlink to throw a help event. If the audio group is not specified, a help event will trigger the nomatch audio group to play (or the initial audio group if nomatch audio group is not specified).

*End*

Name (Label)	Req'd	Max 1	Notes
<b>yes_audio_group</b> (Yes)	No	Yes	Played when the caller chose the "yes" option. If not present, no audio will play when this option is chosen.

**Studio Element Folder:** Menu

**Class Name:** com.audium.server.voiceElement.menu.MYesNoMenu