



Nimblevox Build User's Guide

Document Number: B00002012070901
Revision Number: 1.0.30
Date Created/Updated: 13 Oct 2016

TRADEMARK

This document contains proprietary information that is protected by copyright; it is intended for your internal use only. It is not to be disclosed to third parties. All rights reserved. No part of this document may be photocopied or reproduced in any way without the prior written permission of NewNet Communication Technologies, LLC. (NewNet) The information contained in this document is subject to change without notice. NewNet makes no warranty of any kind with regard to this material. NewNet shall not be liable for errors contained herein or for incidental or consequential damages in connection with the use of this material.

COPYRIGHT INFORMATION

In no event shall the authors or distributors be liable to any party for direct, indirect, special, incidental, or consequential damages arising out of the use of this software, its documentation, or any derivatives thereof, even if the authors have been advised of the possibility of such damage. The authors and distributors specifically disclaim any warranties, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. This software is provided on an "as is" basis, and the authors and distributors have no obligation to provide maintenance, support, updates, enhancements or modifications.

GOVERNMENT USE

If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

ALL RIGHTS RESERVED

Copyright 2011-2020 by:

NewNet Communication Technologies, LLC.

2 Trap Falls Road

Shelton, CT 06484 1-203-926-0049

www.newnet.com

Revision History

Date	Revision	Modification	Modifier
09 July 2012	1.0.0	New Document.	Carrie Powers
26 July 2012	1.0.1	Added Appendix with Installation Instructions.	Carrie Powers
15 August 2012	1.0.2	Updated Record Node. Added Unjoin Node and Conference Node. Updated Service Node.	Variya Ruenprom
19 September 2012	1.0.3	Updated HTTP Node. Updated Edit Project Name screen. Updated Prompt Node screen and added text-to-speech built-in variables. Added Edit Default Language, Edit Default Prompt Group, Edit Default Currency and As digits.	Variya Ruenprom
08 October 2012	1.0.4	Updated Recording node.	Variya Ruenprom
11 October 2012	1.0.5	Updated Addition node to Math node.	Variya Ruenprom
17 October 2012	1.0.6	Removed CDR reference in HTTP Node.	Variya Ruenprom
29 October 2012	1.0.7	Added Date node.	Variya Ruenprom
05 November 2012	1.0.8	Added String node and Billing Log node.	Variya Ruenprom
13 November 2012	1.0.9	Added Write To Log node. Updated various screen captures.	Variya Ruenprom
15 November 2012	1.0.10	Added Javascript node. Updated Prompt node. Updated various screen captures.	Variya Ruenprom
26 November 2012	1.0.11	Updated Section 5 to be the Menu Button instead of just being Deploy and Undeploy.	Brad Martin
30 November 2012	1.0.12	Updated Date node to include the use of current date and time as variables.	Variya Ruenprom
05 December 2012	1.0.13	Updated Settings screen. Added Arguments section to Settings. Added Send Event and Receive Event nodes.	Variya Ruenprom

13 December 2012	1.0.14	Updated two images because the term Dashboard has been retired.	Brad Martin
02 January 2013	1.0.15	Added Email node.	Variya Ruenprom
04 January 2013	1.0.16	Updated Prompt node - custom grammars.	Variya Ruenprom
11 January 2013	1.0.17	Updated Softphone section - added debug log Updated SPOTBuild to nimblevox build	Variya Ruenprom
15 January 2013	1.0.18	Updated Deploy section - added Production checkbox Added Copy/Move/Extract Multiple Nodes	Variya Ruenprom
30 January 2013	1.0.19	Updated Date section - added more to describe the Formatted Result	Brad Martin
06 February 2013	1.0.20	Added Edit to Menu section	Variya Ruenprom
14 February 2013	1.0.21	Added AMD properties Updated Edit Menu	Variya Ruenprom
04 March 2013	1.0.22	Updated AMD mode	Variya Ruenprom
30 April 2013	1.0.23	Updated Reference section Added Build Processes section	Variya Ruenprom
03 July 2013	1.0.24	Updated the Custom VXML node. Added note to Soft Phone section - only supported on Chrome and Firefox.	Carrie Powers
30 July 2013	1.0.25	Added missing images. Updated screens/information for nodes that must apply to a specific connection or conference. Updated Prompt node indicating the custom grammars can also be dynamically referenced.	Carrie Powers
17 Jun 2013	1.0.26	Updated email node to reflect sender changes.	Daniel Walter
11 Jun 2015	1.0.27	Updated with subroutine details.	Daniel Walter
09 Nov 2015	1.0.28	Further subroutine details.	Daniel Walter
09 Nov 2015	1.0.28	Further subroutine details.	Daniel Walter
11 Aug 2016	1.0.29	Updated prompt node clarifying collection variables. Updated condition node for javascript function usage.	Corey Geesey

13 Oct 2016	1.0.30	Updated prompt node with new TTY option.	Corey Geesey
-------------	--------	--	--------------

Table of Contents

1	Introduction	8
1.1	Identification	8
1.2	Supported Browsers	8
1.3	System Overview	8
1.4	Document Overview	8
1.5	References	8
2	Project Screen Layout and Features	9
3	Canvas Screen Layout and Features	11
3.1	Move Canvas	11
3.2	Adjust Display Size of Nodes	12
3.3	Pages	12
3.3.1	View Page	13
3.3.2	Create New Page	13
3.3.3	Delete Page	13
3.3.4	Edit Page Name	13
3.3.5	Reorganize Pages	14
4	Nodes	15
4.1	Prompt	15
4.1.1	Text-to-Speech	15
4.1.2	Audio file	16
4.1.3	Audio URL	16
4.1.4	Recording	17
4.1.5	System Audio	17
4.1.6	TTY/TDD	17
4.2	Record	19
4.3	Play Tone	23
4.4	DTMF Detect	24
4.5	Custom VXML	25
4.6	Answering Machine Detect	27
4.7	Assign Variables	32
4.8	Switch	32
4.9	Condition	33
4.10	Increment Variable	34
4.11	Math	35
4.12	Date	35
4.13	String	36
4.14	Timer	37
4.15	Javascript	37
4.16	Start Service	38
4.17	Start Subroutine	39
4.18	Import Subroutine	40
4.19	Exit Subroutine	41
4.20	Incoming Call	41
4.21	Outdial	42
4.22	Join Connections	43
4.23	Unjoin	44
4.24	Conference	45
4.25	Disconnect	46
4.26	Email	46
4.27	HTTP Request	47
4.28	Send Event	50

4.29 Receive Event	51
4.30 Billing Log	52
4.31 Comment	53
4.32 Go To	53
4.33 Target	54
4.34 Write to Log	54
4.35 Fork	55
4.36 End	55
5 Create Application	56
5.1 Add Node	56
5.2 Configure Node	56
5.2.1 Notes	57
5.2.2 Syntax Help	58
5.3 Delete Node	59
5.4 Clone Node	60
5.5 Copy/Move/Extract Multiple nodes	60
5.6 Move Node	62
5.7 Connect Nodes	62
5.8 Return to Connection Source	63
6 Menu Button	65
6.1 Deploy	65
6.1.1 Deploy Application	65
6.1.2 Remove Application from Number or Service	66
6.2 Edit	67
6.3 Settings	67
6.3.1 Settings	68
6.3.1.1 Edit Project Name	68
6.3.1.2 Edit Default TTS Language	68
6.3.1.3 Edit Default ASR Language	68
6.3.1.4 Edit Default Prompt Group	68
6.3.1.5 Edit Default Currency	68
6.3.1.6 Edit Project Log Level	68
6.3.2 Arguments	68
6.4 Prompts	69
6.5 Share/Export	72
6.6 Soft Phone	72
7 Build Processes	74

1 Introduction

1.1 Identification

This document is for administrators of the Nimblevox Build system.

1.2 Supported Browsers

Nimblevox Build can be used on any browser.

1.3 System Overview

Nimblevox Build is a powerful graphical development environment that allows anyone to create cutting edge Voice and Speech applications. These applications are put together by interconnecting Nodes which represent actual VXML and CCXML syntax. Build's easy to use drag and drop interface means a developer can go from idea to implementation in a short time frame. Whether you are building the simple "Hello World" program or a complex "Conference" production application, Nimblevox Build will generate all the necessary syntax for you.

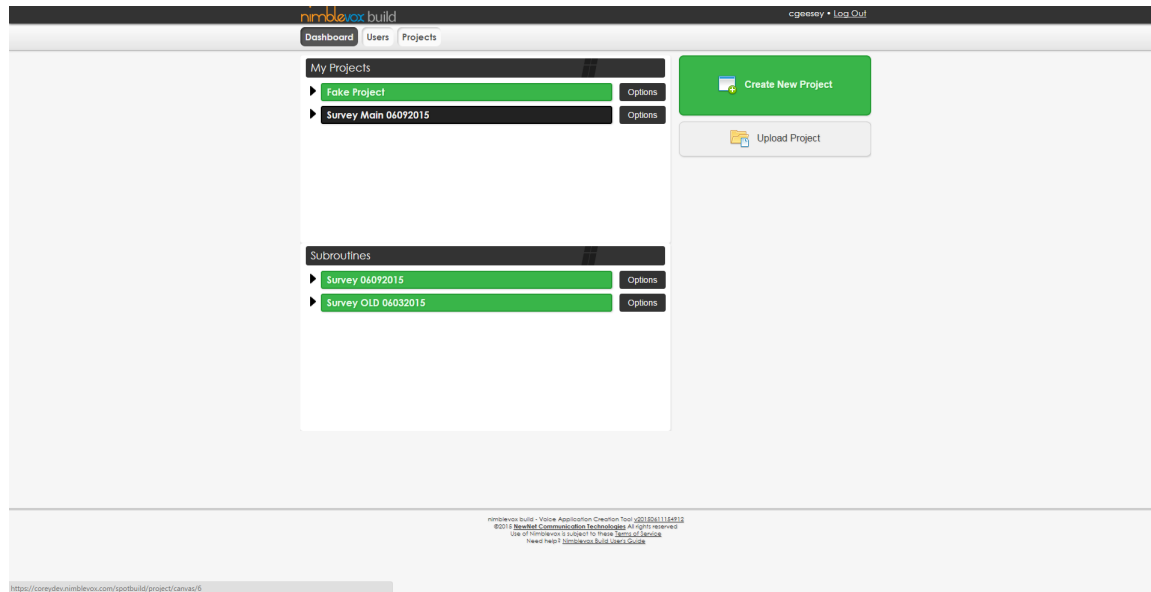
1.4 Document Overview

This document contains screen shots and instructions on how to use Build.

1.5 References

Nimblevox Engine Advisory Reference
Nimblevox Engine Application Developer Reference
Nimblevox Engine Console Guide

2 Project Screen Layout and Features

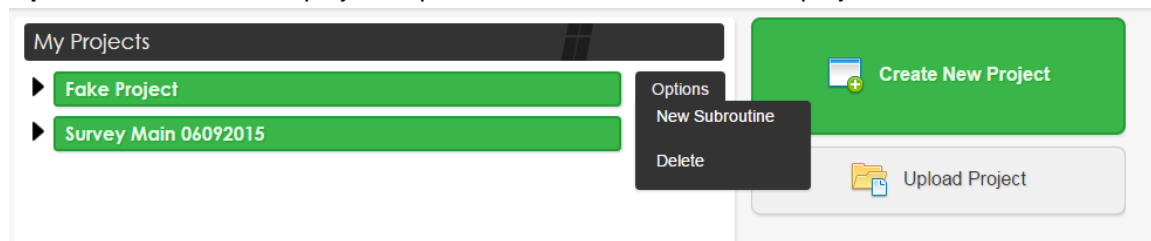


Dashboard -Click this button to return to Dashboard to see My Projects (*listing of available projects*).

Create New Project -Click this button to generate a new application.

Upload Project -Click this to upload an exported project.

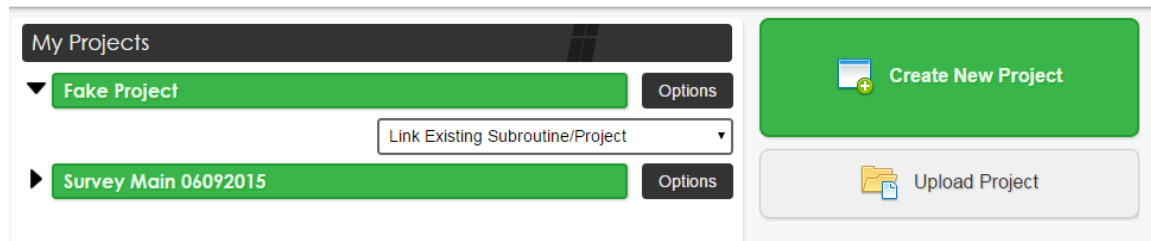
Options -Click this to display the options available for the selected project.



New Subroutine -Create a new subroutine and link it to the selected project.

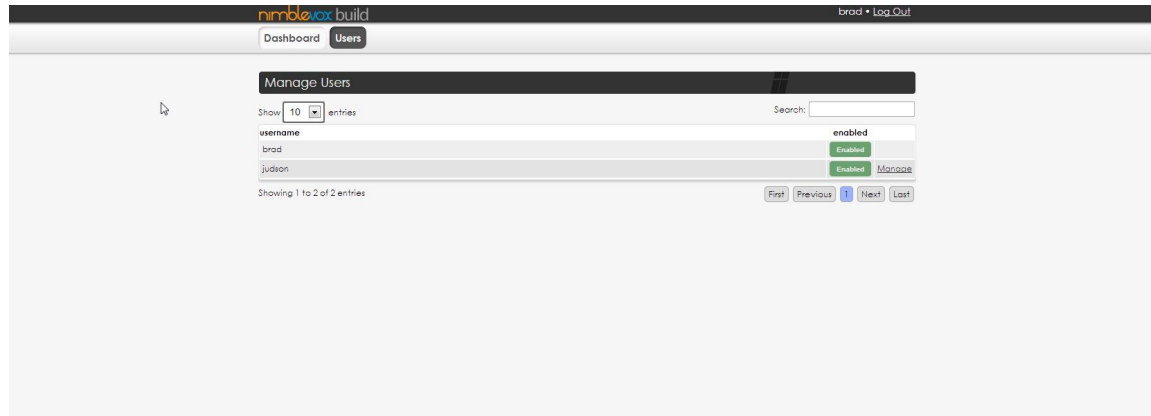
Delete -Delete the selected project.

Arrow (image) -Display a select menu with applicable projects to link as a subroutine.



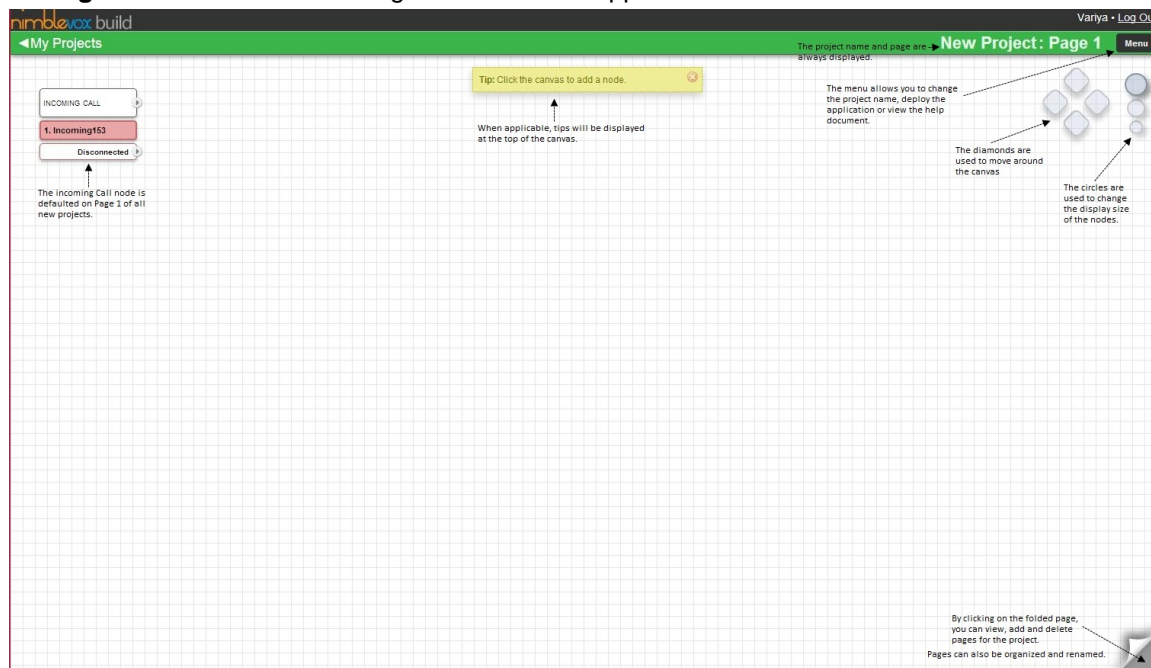
Users - Click this button to see a list of Users.

Project Screen Layout and Features



Enabled/Disabled - Click this button to enable or disable a build user.

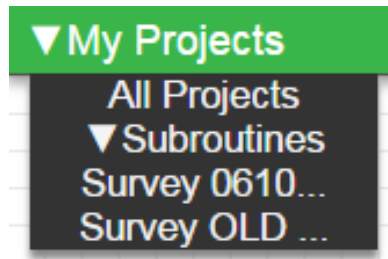
Manage - Click this link to manage a user's Build applications.



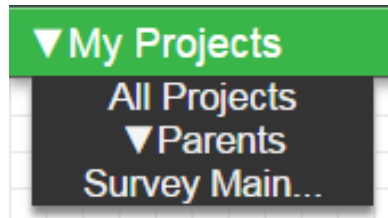
3 Canvas Screen Layout and Features

My Projects -Click this link to display a submenu to navigate to other projects.

Main Project



Subroutine



All Projects -Click this to return to the project dashboard.

Parents -Display a list of projects that have linked this as a subroutine.

Subroutines -Display a list of linked subroutines.

Project Name: Page # - The name of the project and page number are always displayed.

Menu - The menu allows you to deploy the application, change project settings, change prompt groups, download project or view the help document.

Diamonds - The four diamonds in the upper right hand corner of the canvas are used to move the canvas. This is helpful for larger projects with many nodes.

Circles - The three circles in the upper right hand corner of the canvas are used to change the display size of the nodes (*small, medium and large*).

Folded Page - By clicking on the folded page in the bottom right hand corner of the canvas, you can view, add and delete pages for the project. Pages can also be reorganized and renamed.

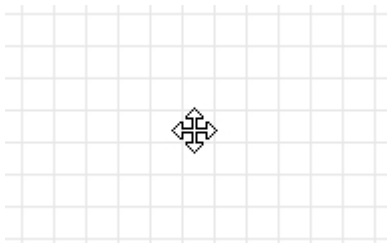
Tips -When applicable, tips will be displayed at the top of the canvas.

3.1 Move Canvas

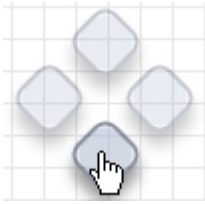
Note: This is helpful when viewing a project with lots of nodes.

To move the canvas so that you can see all of the nodes, you can either:

- Hold down the left mouse button and drag the mouse in the direction you wish to move the canvas.



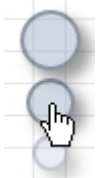
- Click on one of the diamonds in the upper right hand corner to move the canvas up/down/right/left.



3.2 Adjust Display Size of Nodes

Note: This is helpful when viewing a project with lots of nodes.

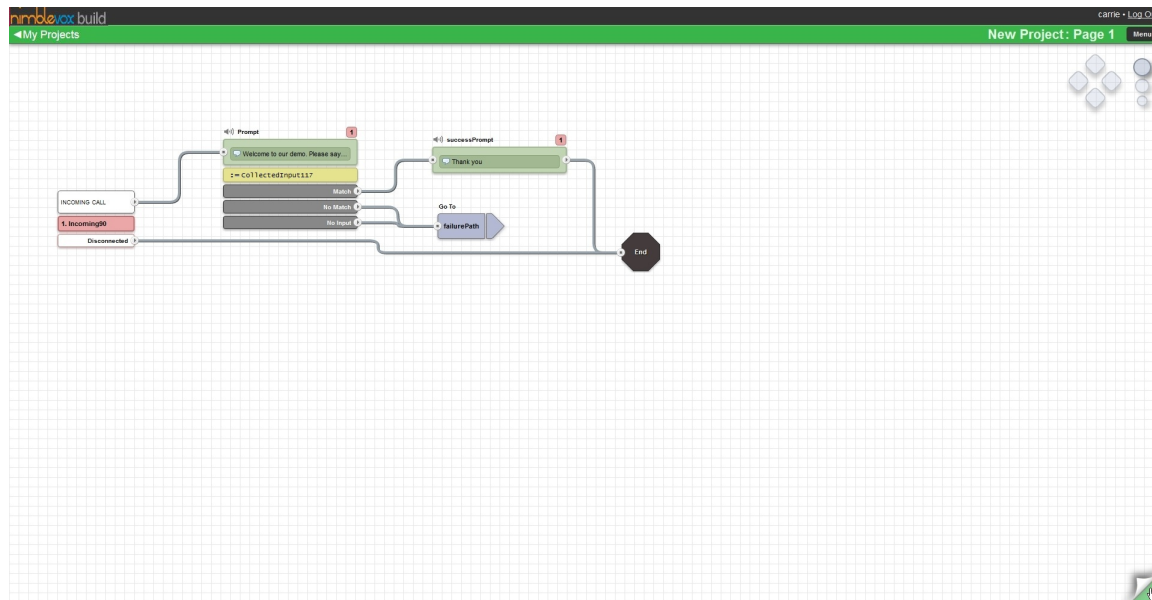
In the upper right hand corner, there are three circles (*small, medium or large*). The circle corresponding to the current size is highlighted. To adjust the node size, click on one of the circles.



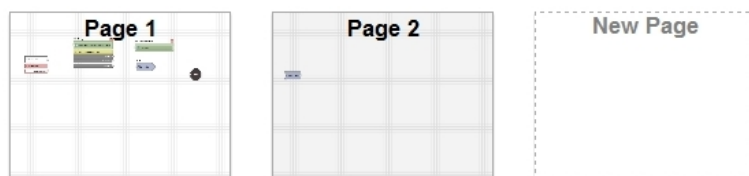
All nodes on the canvas are resized.

3.3 Pages

To view all pages within an application call flow, click the folded page icon in the bottom right hand corner of the screen.

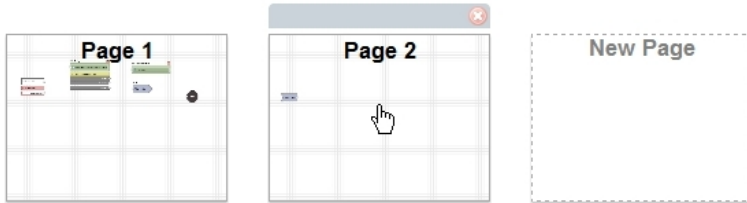


This will display all pages within the application call flow.



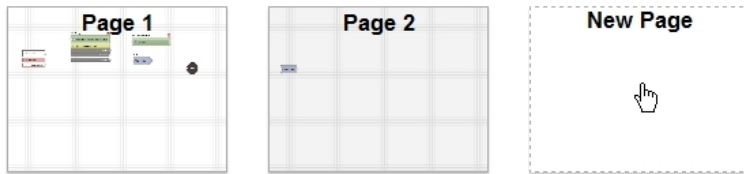
3.3.1 View Page

To view a specific page, simply click that page and it is displayed.




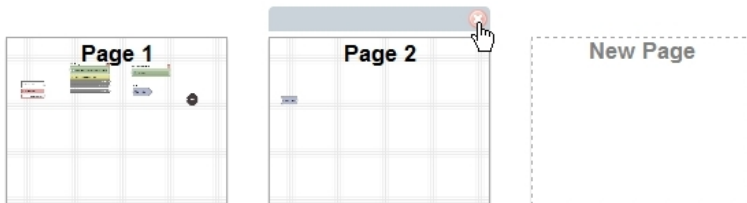
3.3.2 Create New Page


Click the page titled 'New Page' and a new page is created and added to the application call flow.

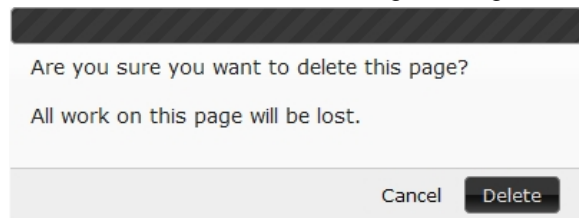


3.3.3 Delete Page

To delete a page, move the mouse over the node until the  button is displayed in the upper right hand corner.



Click the  button and a warning message is displayed.

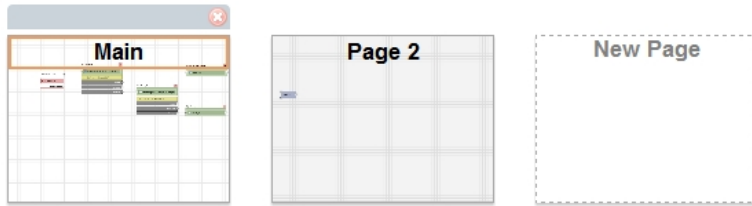


Click the  Delete button and the page is removed.


Note: If there is only one page, it cannot be deleted.

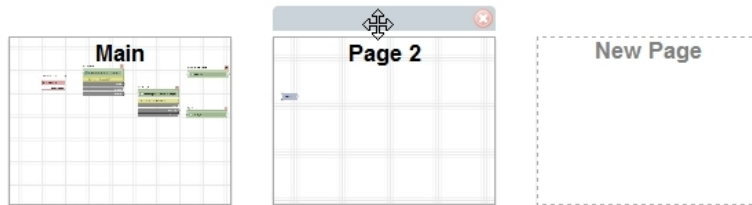
3.3.4 Edit Page Name

To edit the name of the page, click on the current description and then type in the new description.



3.3.5 Reorganize Pages

To reorganize the pages, move the mouse over the page until the  cursor is displayed.



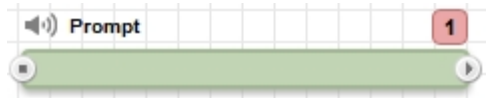
While holding the left mouse button, drag the page to the appropriate location.

4 Nodes

The following are the current available nodes:

4.1 Prompt

The Prompt node plays prompt(s) and collects input(s).



- A prompt can be marked as interruptible.
- Prompts can be text-to-speech, an audio file, an audio URL, a recorded message, system audio or teletype text (TTY).

4.1.1 Text-to-Speech

Text-to-speech prompts support built-in playback of certain variables. Example:

- Your current balance is `asCurrency(@=CurrencyVariable@)`.

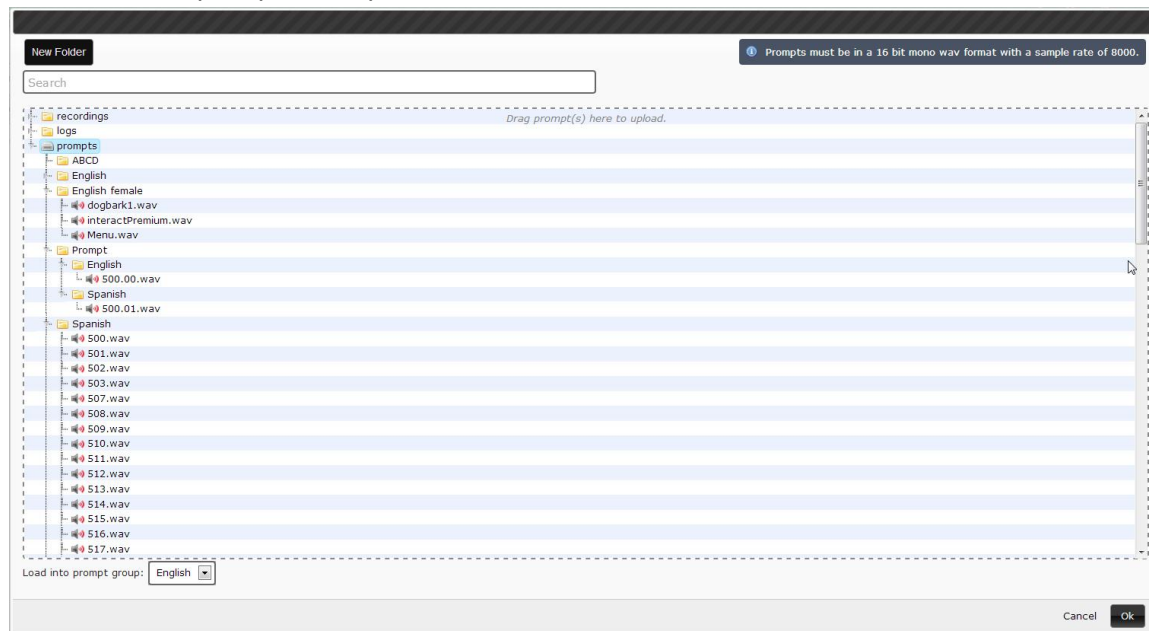
By using the `asCurrency()` tag, the engine is instructed to treat the value in `CurrencyVariable` as a monetary value, speaking it accordingly.

Values provided to playback tags must be specifically formatted. See each tag below for formatting instructions for each playback type.

- **asDate(value)** - Value must be formatted as yyyyymmdd. If a component is not available, it must be replaced with ? characters.
- **asTime(value)** - Value must be formatted as hhmmx (hh = hours, mm = minutes, x = meridian). "x" maybe one of: a- AM or p - PM.
- **asCurrency(value,[type])** - Value must be formatted as mm.nn and type must be formatted as uuu (ISO4217). If not provided, the project default currency will be used.
- **asBoolean(value)** - Value must be true or false.
- **asDigits(value, value)** - The first value must be one or more digits (0-9). Digits will be spoken back as discrete value. The second value is the milliseconds to wait between playing digits.
- **asNumber(value)** - Value must be one or more digits (0-9) with an optional decimal point (.) and optional leading plus (+) or minus (-) sign.
- **asPhone(value)** - Value must be string of digits (0-9) with an optional extension indicator (x) and extension.

4.1.2 Audio file

You can record prompts and upload them.



- Simply click and drag the prompt you want to upload to the designated area on the screen.
- Select the prompt group you want to upload the prompt to. Prompt group is used for multiple languages.

4.1.3 Audio URL

You can specified the URL for prompt to play.



- Enter the URL path or use variable.

4.1.4 Recording

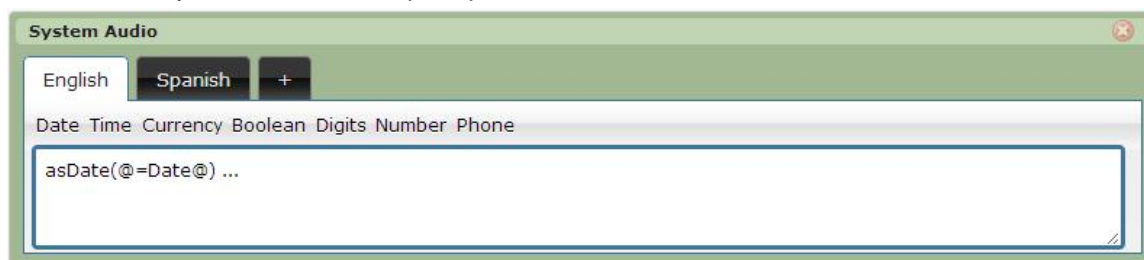
You can play recorded prompt.



- Enter the recording path or use the variable.

4.1.5 System Audio

You can use system audio file for prompt.



To upload system prompt, go to the Menu, click on Prompts and select System tab.



Drag and drop the file on the screen. Uploaded file will be in green.

4.1.6 TTY/TDD

You can play text as baudot tones for teletype/telecommunications device for the deaf machines.



- Enter the text you would like converted into baudot tones.
- Inputs can be collected in DTMF, Voice, Both, or Dynamically (by configuring a variable for Custom Input Mode and using the Dynamic Input Mode) both using either built-in or custom grammars.

Configure Prompt

Prompt Collect **Connection**

Collection Grammar(s)

Built-In: Digits Mode: Both

Timeout (millis): 4000

Custom Input Mode: None @=

Custom DTMF: None @= Select

Custom Voice: None @= Select

DTMF Properties

Inter-digit Timeout (millis): 4000

Termination Timeout (millis): 0

Termination Key: #

Voice Properties

Confidence Level (0-1): 0.5

Complete Timeout (millis): 1000

Incomplete Timeout (millis): 3000

Max Speech Time (millis): 4000

Digits Properties

Min Length: 1 @= Max Length: 1 @=

Assign Variables

Input: CollectedInput967 := [Input] ⓘ

Optional Routes

Match ☐ No Match ☐ No Input ☒ Max Speech Time ☐

Cancel Ok

- The user can either dynamically reference custom grammars (@=variable@) or upload a custom grammar file using the select button, and then dragging the file to the designated area to upload. Uploaded grammar files reside in /interact/users/<userId>/public.

New Folder

Search

grammars

- dtmf_dest.xml
- voice_dest.xml

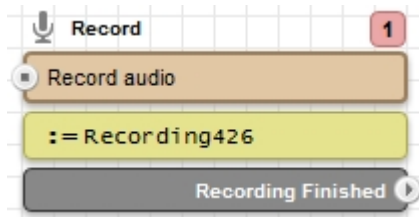
Drag file(s) here to upload.

Cancel Ok

- Optional routes based on the collected inputs: 'Match', 'No Match', 'No Input' or 'Max Speech Time'.
Note: 'Max Speech Time' is only available when Mode='Voice' or 'Both'.
- Collection variable properties are as follows:
 - utterance - The raw string of words that were recognized. In the case of a DTMF grammar, this variable will contain the matched digit string.
 - interpretation - The interpretation of the user input
 - confidence - A value in the range 0.0-1.0. A value of 0.0 indicates minimum confidence of utterance, and a value of 1.0 indicates maximum confidence of utterance.
 - inputmode - The mode in which the user input was provided: dtmf or voice
 - status - A value in the list: match, nomatch, noinput.
 - error - If not empty, a string defining the error that occurred during collection.
- Must apply to a specific connection or conference.

4.2 Record

The Record node records audio input, storing it into a variable.



- Can play a beep prior to recording or allow DTMF to terminate recording.
- Can specify a silence timeout or max record time.
- May include the optional route of 'No Audio Detected'.

Configure Record

Record Storage Connection

Label

Details

Play Beep ☒ Yes ☐ No

Allow DTMF Terminate ☒ Yes ☐ No

Silence Timeout
(1-3600 seconds)

Max Record Time
(1-7200 seconds)

Assign Variables

Recording := ⓘ

Optional Routes

ⓘ If "No Audio Detected" is not routed, the "Recording Finished" route will be taken.

No Audio Detected ☐

Cancel Ok

- Can specify where to store the recorded file. Unless specified, the file location is defaulted to the recordings directory. If the directory you specify does not exist it will be created.

Configure Record

Record Storage Connection

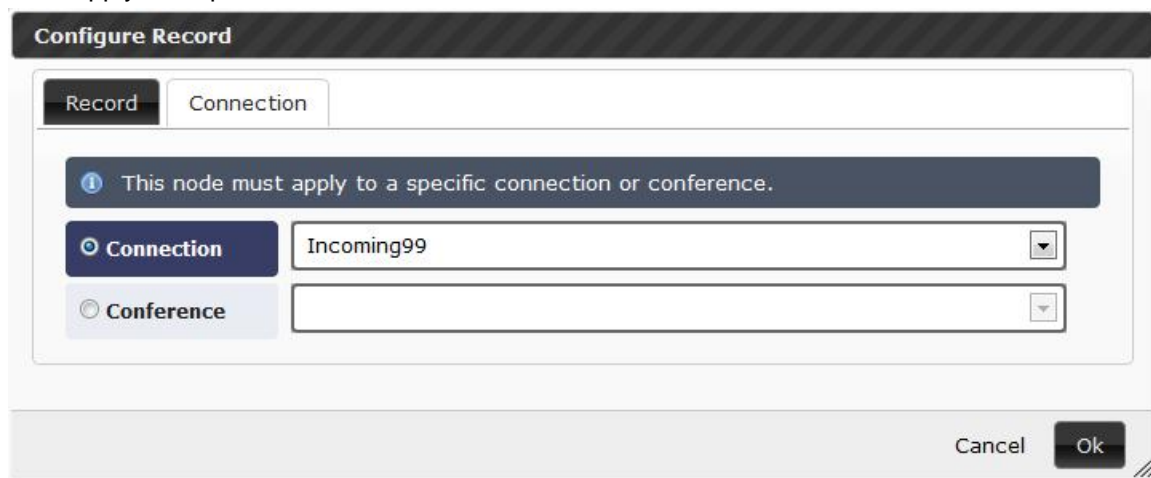
ⓘ Please specify information about the recorded file.

File Name ⓘ

File Location ⓘ

Cancel Ok

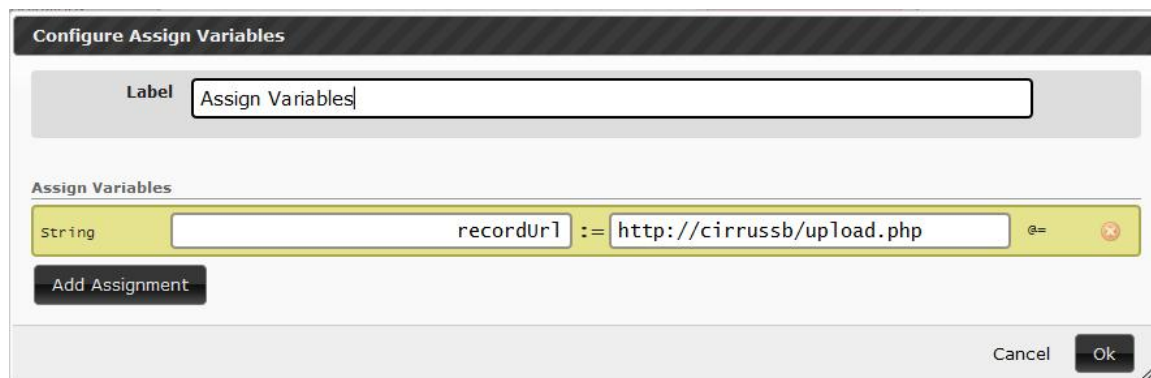
- Must apply to a specific connection or conference.



The 'Configure Record' dialog box has two tabs: 'Record' and 'Connection'. The 'Record' tab is active. It contains a message: 'This node must apply to a specific connection or conference.' Below this, there are two radio buttons: 'Connection' (selected) and 'Conference'. The 'Connection' radio button is associated with a dropdown menu showing 'Incoming99'. The 'Conference' radio button is associated with an empty dropdown menu. At the bottom right, there are 'Cancel' and 'Ok' buttons.

Recording can be used to record a conference or outdial call. You can use HTTP to Post a recorded audio file to a webserver specified by a developer. There are two ways to accomplish this: Assign Variables Node and HTTP Node.

The Assign Variable Node is used to designate a URL where the recording should be posted. The reserved variable "recordUrl" should be given the value of the URL. The value of the assign variable is the webserver location. This node must be created before the record node.



The 'Configure Assign Variables' dialog box has a 'Label' field containing 'Assign Variables'. Below this, there is a section titled 'Assign Variables'. It contains a row with a yellow background: 'String' in a dropdown, followed by a text field containing 'recordUrl', an equals sign, and another text field containing 'http://cirrussb/upload.php'. To the right of this row are icons for adding and removing items. Below this row is an 'Add Assignment' button. At the bottom right, there are 'Cancel' and 'Ok' buttons.

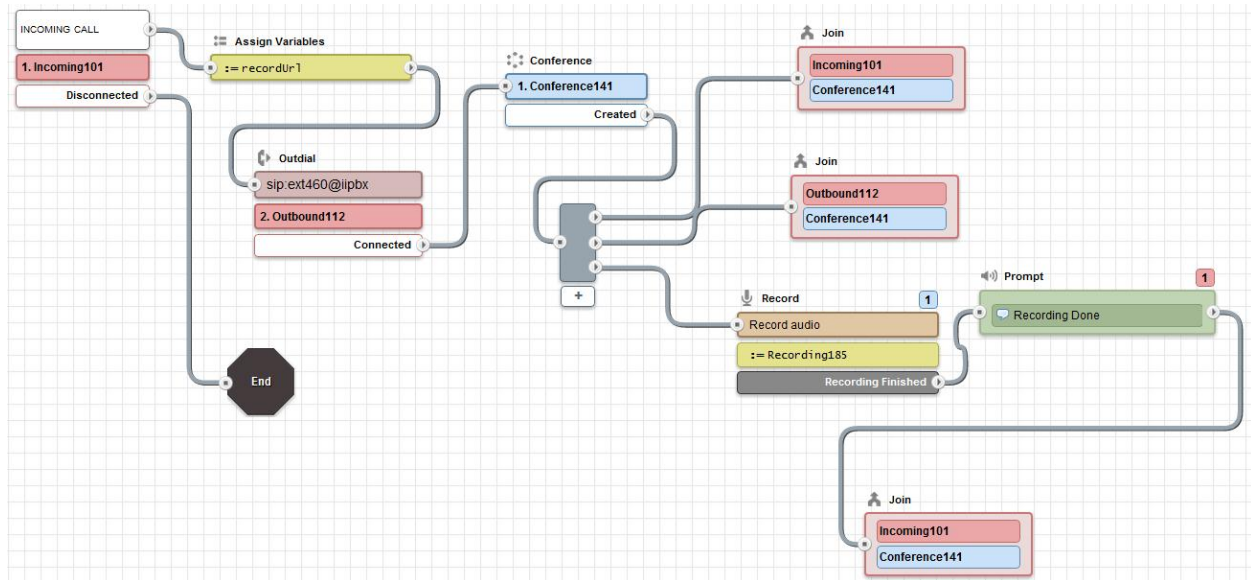
The following will be sent with the HTTP Post:

- **Recording** - Recorded audio file. (File)
- **SessionId** - Unique Id of current call. (String)
- **Size** - File size in bytes. (String)
- **maxTime** - Value is either 'true' or 'false'. Indicates if recording reached max time limit. (String)
- **duration** - Length of recording in milliseconds. (String)
- **termChar** - Key press used to terminate recording. (if available) (String)

The receiving server MUST respond with Valid xml.

```
<?xml version="1.0"?>
<postAudio>
<response>[success or failure]</response>
<file>[http accessible file path]</file>
</postAudio>
```

Below is the example of how assign variable node can be configured in a project.



Using HTTP Node is another way to post http to a file.

Configure HTTP Request

Details

Headers

Auth

Response

Label

HTTP Request

Method

POST

URL

http://cirrussb/upload.php

@=

Content-Type

Multipart Form (multipart/form-data)

Charset

utf-8

May be overridden by Content-Type on the Headers tab.
Defaults to *application/json* if not set.

Default is *utf-8*.

Parameters (sent in request body)

Name	Value
name	Recording185
sessionId	@=Incoming101.id@
destDir	sandBox/upload

Add Parameter

Timeout (milliseconds)

3000

If reached, the **Connection Error** route will be taken.

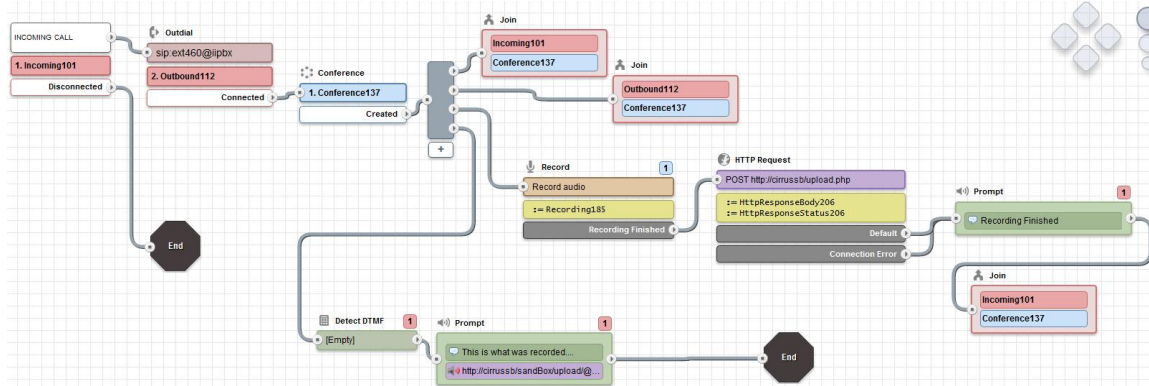
Cancel

Ok

- Select 'Post' method.
- Enter the webserver location in the URL.
- Select 'Multipart Form' content-type.

- Enter the parameter(s) that need to be sent to the http.

Below is the example of how the HTTP node can be configured in the project.



4.3 Play Tone

The Play Tone node plays the DTMF tone equivalent of the specified values.



- Each key tone or pause specified will be played back for the specified duration.
- A comma (",") may be used to insert a pause.

Example: Could be used with an outdial application call flow to dial an extension.

Configure Play Tone

Play Tone **Connection**

Label

Details

i Each key tone or pause specified in the "Tone(s)" field will be played back for the specified duration. A comma (",") may be used to insert a pause. For example:
 12*,,4
 With a Tone Duration of 100 and Pause Duration of 500 will:
 Play tones 1, 2, and * for 100 ms each
 Pause for 1000 ms
 Play tone 4 for 100 ms

Tone(s) @=

Tone Duration
 (milliseconds)

Pause Duration
 (milliseconds)

Cancel **Ok**

- Must apply to a specific connection.

Configure Play Tone

Play Tone **Connection**

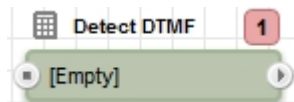
i This node must apply to a specific connection.

Connection

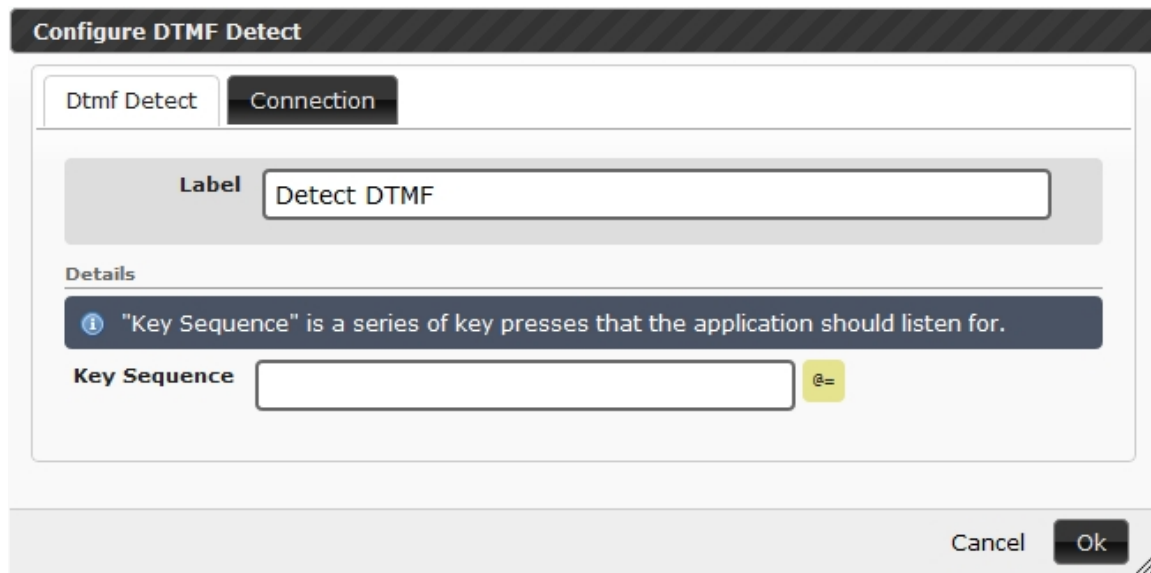
Cancel **Ok**

4.4 DTMF Detect

The DTMF node detects specified DTMF keys.



- Specified series of key presses that the application should listen for.



Configure DTMF Detect

Dtmf Detect | Connection

Label: Detect DTMF

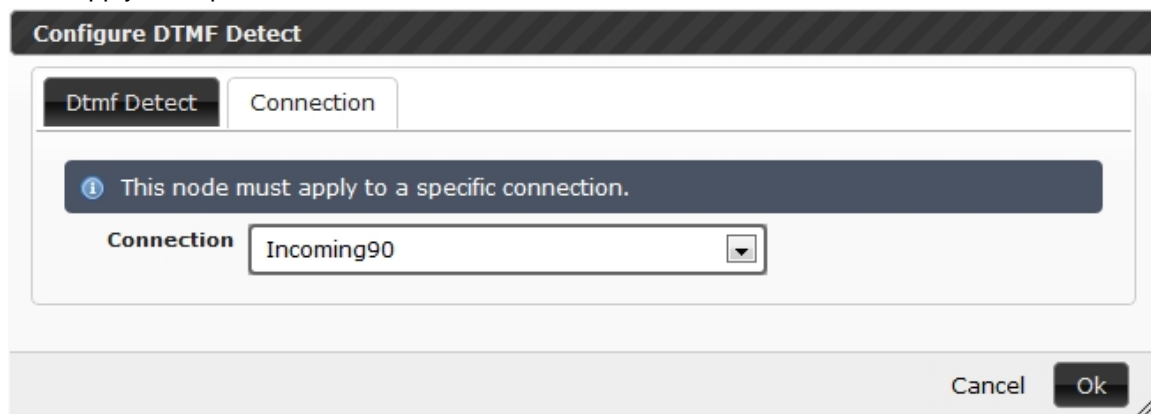
Details

"Key Sequence" is a series of key presses that the application should listen for.

Key Sequence: [] @=

Cancel | Ok

- Must apply to a specific connection.



Configure DTMF Detect

Dtmf Detect | Connection

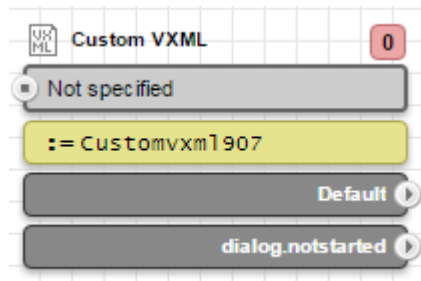
This node must apply to a specific connection.

Connection: Incoming90 [v]

Cancel | Ok

4.5 Custom VXML

The Custom VXML node allows custom VXML to be executed.



Custom VXML 0

Not specified

:= Customvxml1907

Default

dialog.notstarted

- The VXML file can be:
 - Accessed via HTTP
Format: `http://serverName[:port]/pathToFile.vxml`
 - Uploaded using Select button

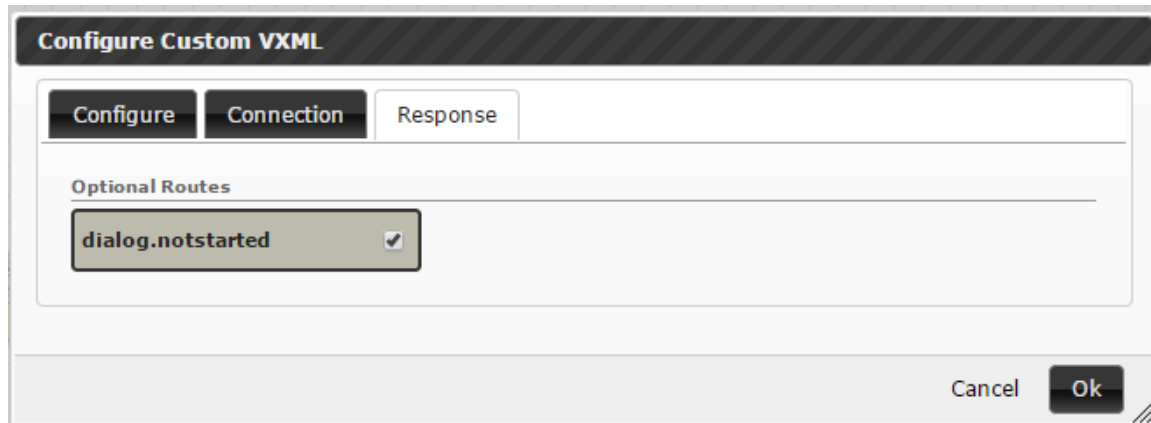
- A variable can be assigned to the custom VXML.

The screenshot shows the 'Configure Custom VXML' dialog box with the 'Configure' tab selected. The 'Label' field contains 'Custom VXML'. The 'VXML File' section has an information box stating: 'VXML file can be: 1. Accessed via http (http://serverName[:port]/pathToFile.vxml) 2. Uploaded using Select button'. Below this is an empty text field and a 'Select' button. The 'Assign Variables' section shows a variable assignment: 'Customvxml' is assigned to 'Customvxml1907' with the value '[Customvxml]'. The dialog has 'Cancel' and 'Ok' buttons at the bottom right.

- Must apply to a specific connection.

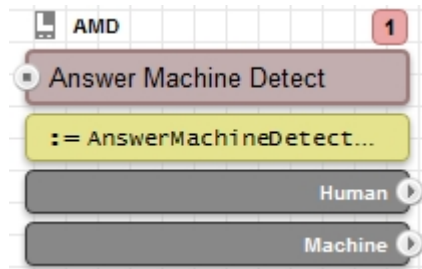
The screenshot shows the 'Configure Custom VXML' dialog box with the 'Connection' tab selected. An information box states: 'This node must apply to a specific connection.' Below this, the 'Connection' dropdown menu is set to 'ParentConnection'. The dialog has 'Cancel' and 'Ok' buttons at the bottom right.

- An optional route can be configured for the custom VXML in the event that the dialog fails to start. If not utilized, the application will behave as if it received a dialog.exit event when the dialog fails to start.



4.6 Answering Machine Detect

The AMD node detects the difference between a live human speaking and an answering machine.



- Possible modes are: 'Live Speaker', 'Machine Tone' or Both.
- There are two routes for Live Speaker:
 - **Human Detected** - Detects live human speaking.
 - **Machine Detected** - Detects the answering machine based on the configured timers under the advanced tab.

Configure Answer Machine Detected

Mode **Advanced** Connection

Label

Mode

i Live speaker mode uses energy based detection to distinguish between a Human vs. Machine speaker. Machine Tone uses a pulse based detection to distinguish between Machine Beep, Fax Tone and Special Information Tones (SIT).

☒ **Live Speaker** ☐ Machine Tone ☐ Both

Assign Variables

General := *i*

Optional Routes

Human Detected ☒ Machine Detected ☒

Cancel **Ok**

- There are four routes for Machine Tone:
 - **Machine Beep** - Detect the answering machine beep.
 - **Machine Fax** - Detect the fax tone.
 - **Machine SIT** - Detect the disconnect number tone.
 - **Timeout** - Determine how long the machine tone detect runs.

Configure Answer Machine Detected

Mode **Advanced** **Connection**

Label

Mode

Live speaker mode uses energy based detection to distinguish between a Human vs. Machine speaker. Machine Tone uses a pulse based detection to distinguish between Machine Beep, Fax Tone and Special Information Tones (SIT).

☐ Live Speaker ☒ **Machine Tone** ☐ Both

Assign Variables

General := [i](#)

Optional Routes

☐ Machine Beep ☐ Machine Fax ☐ Machine SIT

☐ Timeout

Cancel **Ok**

- All six routes are available if Both is selected for the mode.

Configure Answer Machine Detected

Mode **Advanced** **Connection**

Label

Mode

Live speaker mode uses energy based detection to distinguish between a Human vs. Machine speaker. Machine Tone uses a pulse based detection to distinguish between Machine Beep, Fax Tone and Special Information Tones (SIT).

☐ Live Speaker ☐ Machine Tone ☒ Both

Assign Variables

General := [?](#)

Optional Routes

Human Detected <input type="checkbox"/>	Machine Detected <input type="checkbox"/>	Machine Beep <input type="checkbox"/>
Machine Fax <input type="checkbox"/>	Machine SIT <input type="checkbox"/>	Timeout <input type="checkbox"/>

Cancel **Ok**

- Each mode, Live Speaker and Machine Tone, have their own properties. *See inline help for properties description.*

Configure Answer Machine Detected

Mode

Advanced

Connection

Machine Tone Properties

Timeout (millis)

60000

Live Speaker Properties

Initial Silence (millis)

5000

Maximum Word Count (number)

5

Greeting Length (millis)

8000

Minimum Word Length (millis)

100

Post Greeting Silence (millis)

1500

Silence Between Words (millis)

50

Machine Final Silence (millis)

1000

Silence Threshold (millis)

256

Total Time (millis)

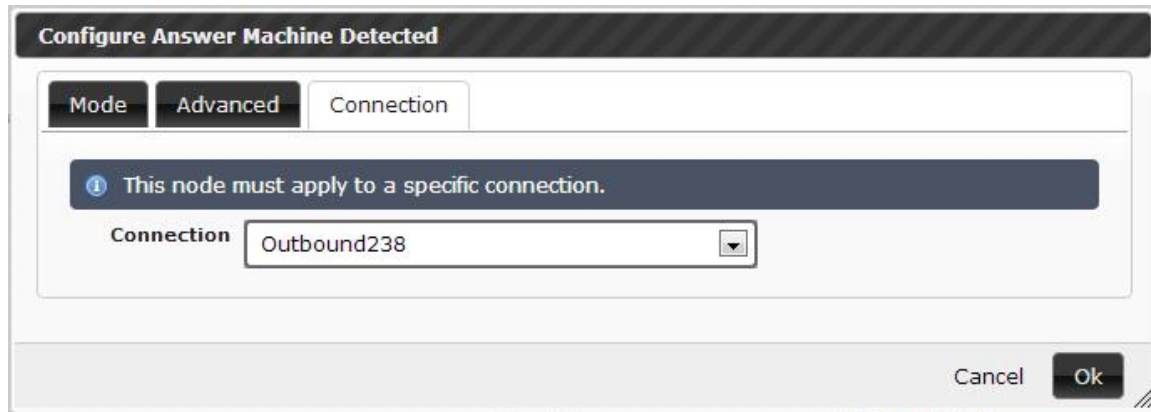
60000

Live Speaker Chart

Cancel

Ok

- Must apply to a specific connection.



Configure Answer Machine Detected

Mode **Advanced** Connection

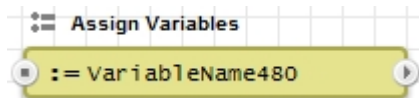
ⓘ This node must apply to a specific connection.

Connection

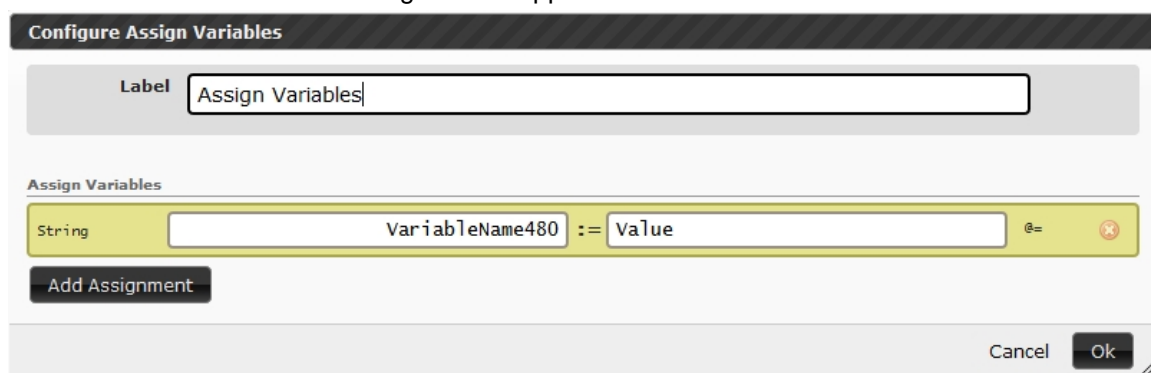
Cancel **Ok**

4.7 Assign Variables

The Assign Variables node sets the value(s) of one or more variables.



- Variables can then be used throughout the application call flow.



Configure Assign Variables

Label

Assign Variables

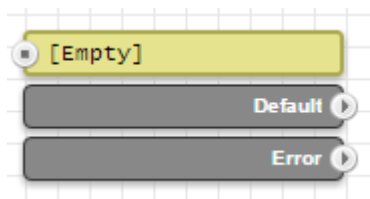
String := @=

Add Assignment

Cancel **Ok**

4.8 Switch

The Switch node performs an equals comparison on a variable, allowing routes for configurable matching values.



- The variable specified will be compared for equality with the list of conditions configured.
- Each condition will create a route-able edge for this node.

Configure Switch

Label

Variable

Conditions

The variable above will be compared for equality with the list of conditions you configure below. Each condition will create a routable edge for this node.

=

Add

Optional Routes

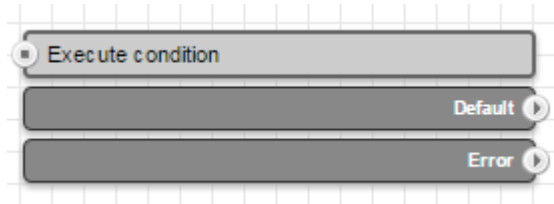
Error ☒

Cancel **Ok**

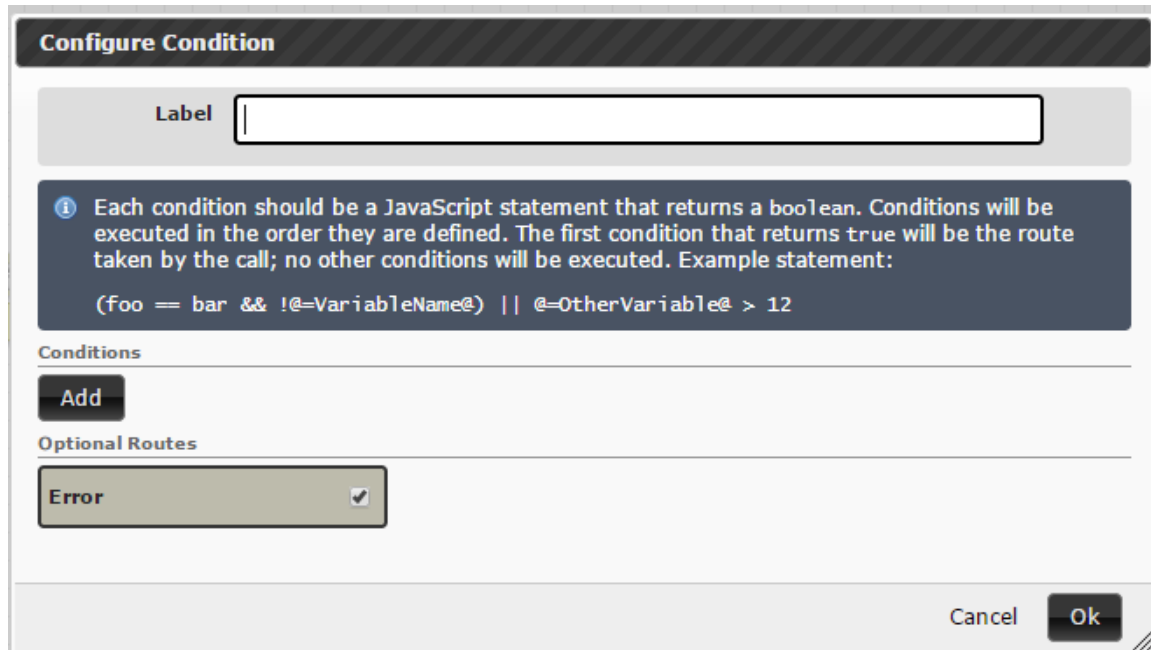
- A 'Default' route is also available.
- An optional 'Error' route is also available. If checked, a new port will be available on the Switch Node widget. If an error occurs during execution of the conditions, the application will take the error route. If not selected, the application will take the default route in the event of an error.

4.9 Condition

The Condition node evaluates Boolean expressions allowing for conditional routes.



- Conditions will be executed in the order they are defined.
- The first condition that returns 'true' will be the route taken by the call; no other conditions will be executed.



Configure Condition

Label

Each condition should be a JavaScript statement that returns a boolean. Conditions will be executed in the order they are defined. The first condition that returns true will be the route taken by the call; no other conditions will be executed. Example statement:

`(foo == bar && !@=VariableName@) || @=OtherVariable@ > 12`

Conditions

Add

Optional Routes

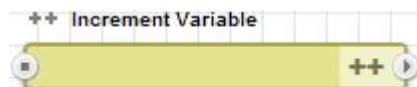
Error ☒

Cancel **Ok**

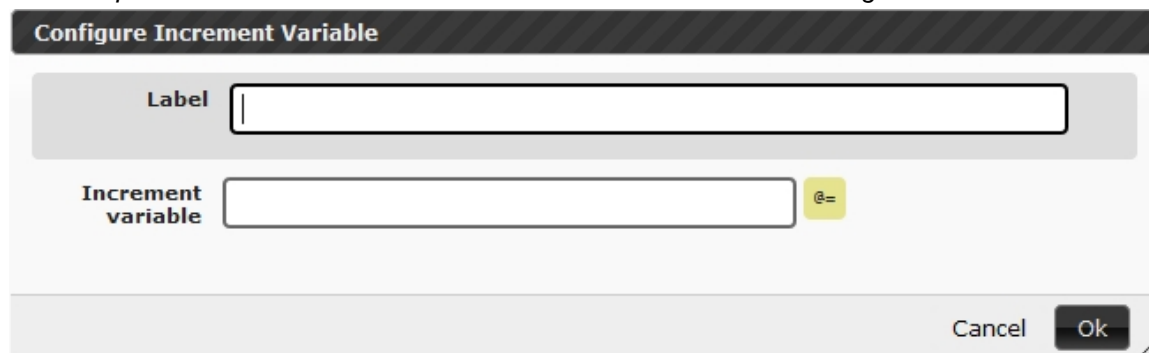
- Built-in javascript functions can be used within the routes. ex: `@=variable.length@` or `@=variable@.length`
For integer comparisons, use `parseInt()` or `parseFloat()` ex: `parseInt(@=variable@) == 1`
- A 'Default' route is also available.
- An optional 'Error' route is also available. If checked, a new port will be available on the Switch Node widget. If an error occurs during execution of the conditions, the application will take the error route. If not selected, the application will take the default route in the event of an error.

4.10 Increment Variable

The Increment Variable node increments (*adds one to*) a variable.



Example: Can be used to increment the number of errors made during a call.



Configure Increment Variable

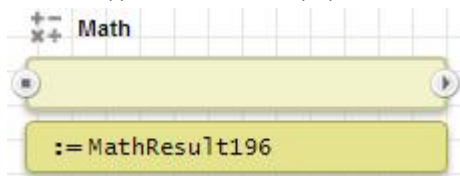
Label

Increment variable **@=**

Cancel **Ok**

4.11 Math

The Math node performs mathematical functionality such as addition (+), subtraction (-), multiplication (*), division (/) and modulus (%).



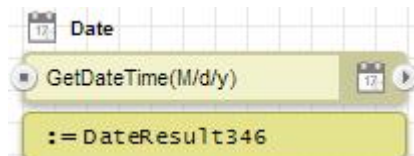
- The values may be a variable containing a number (*delimited with @=@*) or a literal numeric value (e.g. 5).

- If () is not used, multiplication and division is always evaluated before addition and subtraction.

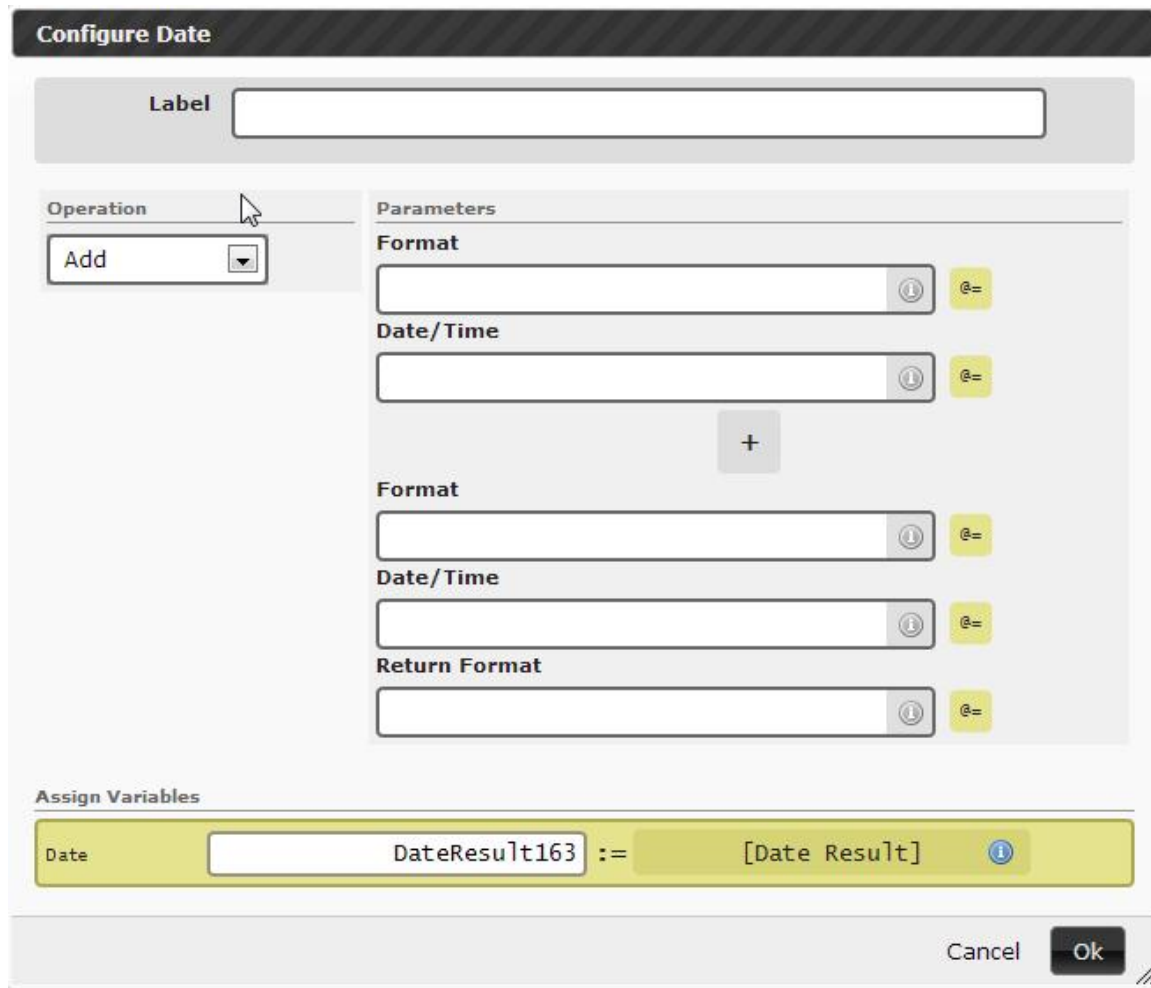
4.12 Date

The Date node performs the following:

- compare dates
- add or subtract hours (*min, sec, days, etc.*) to a date
- get date time (*hh:mm*) from date value



- Date result returns five properties: timestamp, date, time, comparisonResult, and formattedResult.



Configure Date

Label

Operation: Add

Parameters:

- Format: @=
- Date/Time: @=
- +
- Format: @=
- Date/Time: @=
- Return Format: @=

Assign Variables

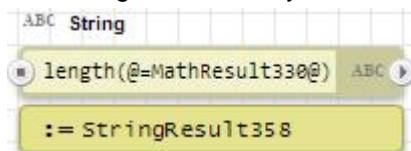
Date: DateResult163 := [Date Result] ⓘ

Cancel Ok

You can have the current date and time assigned to variables and then use those variables throughout your application. Simply select Get Date Time from Operations and enter the format. The date format is yyyyMMDD and the time format is hhmm. The Return Format field can be used to set the Date variable in any format that is specified. For example HH/mm/ss or HH:mm:ss.

4.13 String

The String node allows you to concatenate, split, get length and get substring from a string value.



String

length(@=MathResult330@) ABC ▶

:= StringResult358

- The value may be a literal number or variable.



Configure String Operation

Label

Operation: Split

Parameters:

String ⓘ @=

Delimiter ⓘ @=

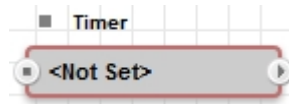
Assign Variables:

General := ⓘ

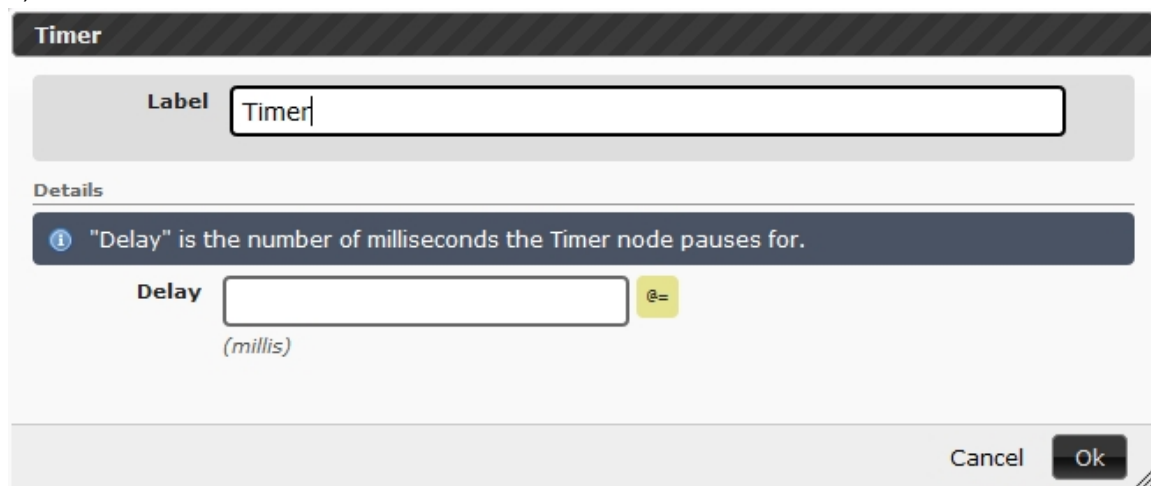
Cancel Ok

4.14 Timer

The Timer node delays for a specified number of milliseconds.



- The value may be a variable containing a number (*delimited with @=@*) or a literal numeric value (e.g. 5).



Timer

Label

Details

ⓘ "Delay" is the number of milliseconds the Timer node pauses for.

Delay ⓘ

(millis)

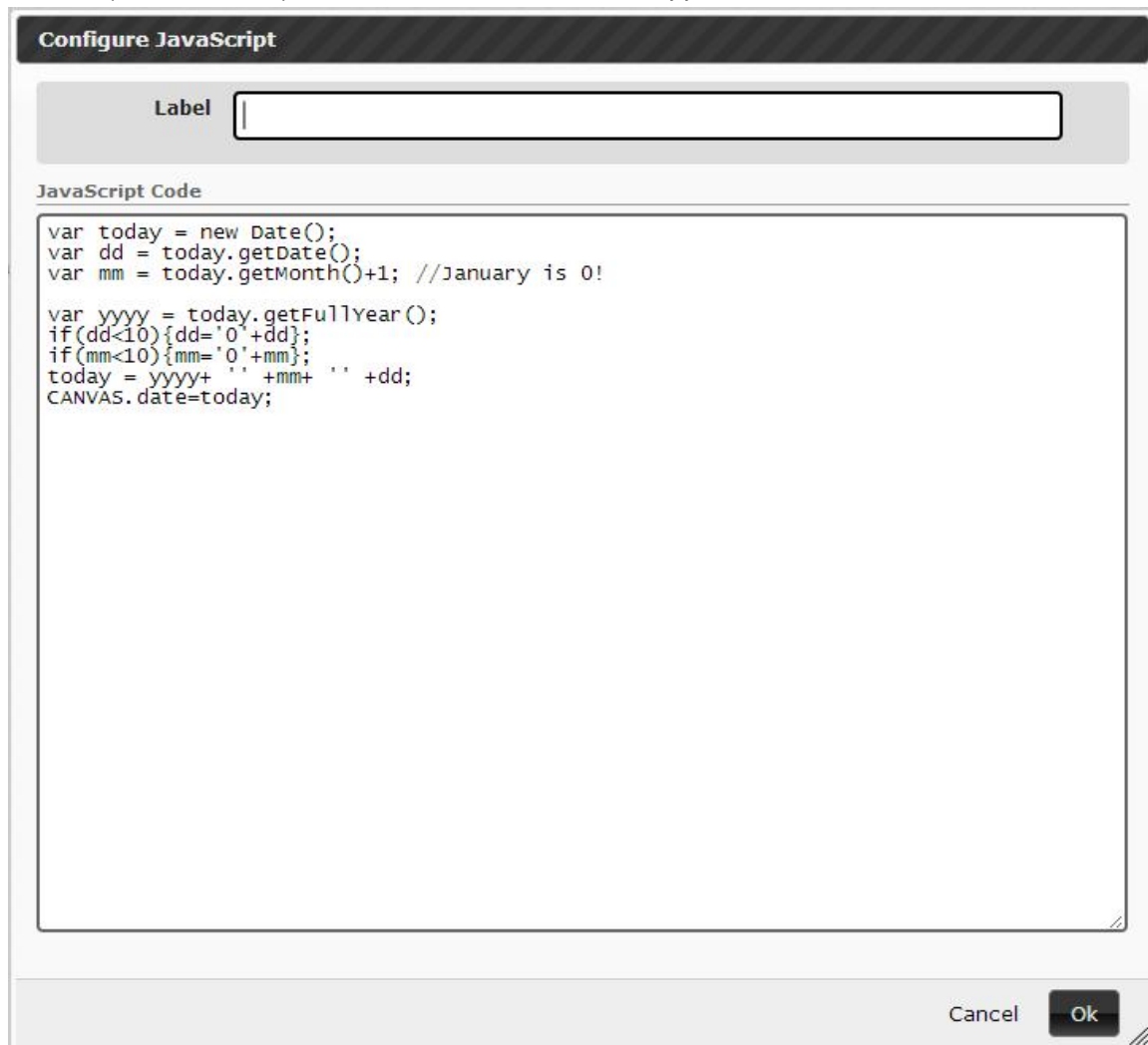
Cancel Ok

4.15 Javascript

The Javascript node allows a user to execute JavaScript.



- `Canvas.(variable name)` is accessible in the rest of the application.

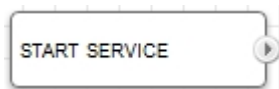


- The node has a limit of 4096 characters.

4.16 Start Service

The Start Service node indicates that the application has been started as a service (e.g. via web or API). This node and/or the Incoming Call node should begin the application call flow.

This node is unavailable to subroutines.



If you need to pass variables to your service, add them to the request string. An example curl command to run:

```
curl -v --request POST
      "https://api.nimblevox.com/nimblevox/api/serviceStart?apiKey=[your
key]&serviceId=[nimblevox build
      service ID associated with your nimblevox build project]&[project variable
name]=[project variable
      value]"
```

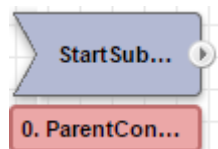
where project variable name can be var1 and project variable value can be 4021234567

Configure the Assign node to access the passed in variables, args.[variable name]. Example:

 A dialog box titled "Configure Assign Variables" with a dark header bar. Below the title bar is a "Label" field containing the text "Assign Variables". Underneath is a section titled "Assign Variables" containing a table with one row. The first column is labeled "String" and contains the text "phonenum". The second column contains the text ":= @args.var1@". To the right of the table is a small circular icon with a right-pointing arrow. Below the table is a button labeled "Add Assignment". At the bottom right of the dialog are "Cancel" and "Ok" buttons.

4.17 Start Subroutine

The Start Subroutine node indicates that the application can serve as a subroutine. (*e.g. via the import node.* This node should begin the subroutine call flow, resuming from the import node in the main project.



The name of the connection inherited from the main project can be assigned a different name by clicking on the node.

Configure Connection

Assign Variables

Connection

ParentConnection

:=

[Connection]

i

Events

i

This node creates a connection that supports the following events. Enabling events below will add them to the canvas. These events are **asynchronous**; they may happen any time during the call.

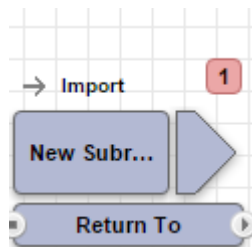
Cancel

Ok

4.18 Import Subroutine

The Import Subroutine node serves to move control of the application over to the indicated subroutine. The subroutine will inherit the indicated connection and have access to all of the parent project's data through the global variable. The import node will link the previous node with the Start Subroutine Node contained within the subroutine. A shortcut to traverse to the indicated subroutine is provided by clicking on the arrow of the import node. This shortcut is disabled if a subroutine is not selected.

This node is unavailable to subroutines.



Click on the main panel for the import node to configure the linked subroutine and namespace. All values stored within the subroutine will be accessible in the parent app, or other subroutines, by appending the namespace provided to the variable name (ex. valueOne => namespace.valueOne).

Configure Import Subroutine

Subroutine **Connection**

Label

Subroutine

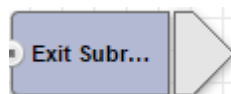
Namespace

To reference variables from within the subroutine, use \$namespace.variableName

Cancel **Ok**

4.19 Exit Subroutine

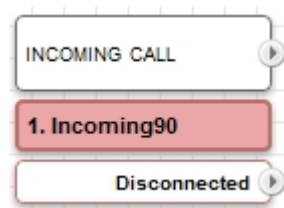
The exit subroutine node signals the end of the subroutine, at which point control will be resumed within the parent app.



4.20 Incoming Call

The Incoming Call node indicates that the application has started via incoming phone call which has been automatically accepted and connected. This node and/or the Start Service node should begin the application call flow.

This node is unavailable to subroutines.



- Creates a connection that supports the following events: 'Connection' and 'Disconnected'.

Configure Connection

Connection Name

Events

i This node creates a connection that supports the following events. Enabling events below will add them to the canvas. These events are **asynchronous**; they may happen any time during the call.

Disconnected ☒

Cancel **Ok**

- Only one may exist per application.

4.21 Outdial

The Outdial node dials an external phone number.

Outdial

[destination]

2. Outbound534

Connected

Disconnected

- Configure the destination information: 'Destination', 'Caller Id' and 'Ring Timeout'.

Configure Outdial

Label

Details

i Destination can be:
 1. Phone number (Digits ONLY)
 2. Properly formed sip URI (sip:user@host[:port]). Port is optional

Destination @=

Caller Id @=

Ring Timeout
 (seconds)

Cancel **Ok**

- Configure the outbound connection information. This node creates a connection that supports the following events: 'Progressing', 'Connected' (*defaulted*), 'Failed', 'Disconnected' (*defaulted*) and 'No Answer'.

Configure Connection

Assign Variables

Connection Outbound275 := [Connection]

Events

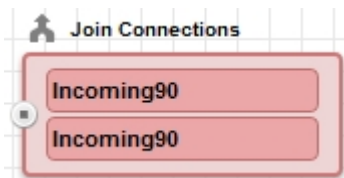
ⓘ This node creates a connection that supports the following events. Enabling events below will add them to the canvas. These events are **asynchronous**; they may happen any time during the call.

Progressing	<input type="checkbox"/>
Connected	<input checked="" type="checkbox"/>
Failed	<input type="checkbox"/>
Disconnected	<input checked="" type="checkbox"/>
No Answer	<input type="checkbox"/>

Cancel **Ok**

4.22 Join Connections

The Join Connections node bridges two call connections or a call connection with a conference.



- Connection 2 can be muted.
- This node supports the following connection events: 'Joined' and 'Failed'.
- These events are asynchronous; they may happen any time during the call.

Configure Join

Label

Join

Connection

Incoming163

Conference

Conference442

Join With

Connection

Incoming163

Mute

Yes

No

Variable

Events

This node supports the following connection events. Enabling events below will add them to the canvas. These events are **asynchronous**; they may happen any time during the call.

Joined

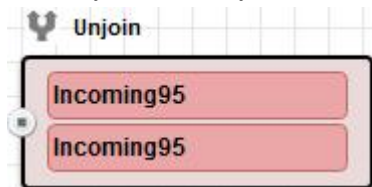
Failed

Cancel

Ok

4.23 Unjoin

The Unjoin node unjoins a connection between two calls or a call and a conference.



- This node supports the following connection events: 'Unjoined' and 'Failed'.

Configure Unjoin

Label

Unjoin

Connection

Incoming163

Unjoin With

Connection

Incoming163

Conference

Conference425

Events

This node supports the following connection events. Enabling events below will add them to the canvas. These events may happen any time during the unjoin.

Unjoined

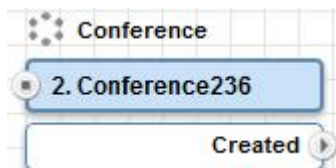
Failed

Cancel

Ok

4.24 Conference

The Conference node allows multiple callers to be on the same call.



- This node supports the following events: 'Created' and 'Destroyed'.

Configure Conference

Label

Name @=

Assign Variables

Conference := ⓘ

Events

ⓘ This node supports the following events. Enabling events below will add them to the canvas. These events are **asynchronous**; they may happen any time during the call.

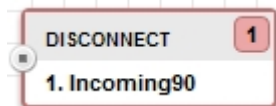
Created ☒

Destroyed ☐

Cancel

4.25 Disconnect

The Disconnect node disconnects a connection.



- Must apply to a specific connection.

Disconnect

Label

ⓘ This node must apply to a specific connection.

Connection ▼

Cancel

4.26 Email

The Email node sends an email to the listed recipient(s).

Send Email

Label

Sender

To Address

Subject

Attachment

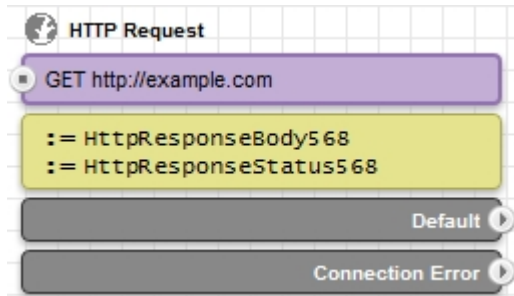
Body

Cancel Ok

- **Sender** - The name the email will appear under. Variables can be used.
- **To Address** - The email address(es). Separate multiple addresses with a comma. Variables can be used.
- **Subject** - The subject of the email. Variables can be used.
- **Attachment** - The file name and path relative to your storage folder containing the attachment. Variables can also be used.
- **Body** - The message to send in the email. Variable can be used.

4.27 HTTP Request

The HTTP Request node makes an HTTP request, returning the response entity into a variable.



- Configure the Method (GET, POST, PUT, DELETE, HEAD, OPTIONS, TRACE, PATCH), URL, Content-Type (e.g. XML (*application/xml*)), charset (e.g. *utf-8*), Body or Parameters and Timeout value (If timeout is reached, the *Connection Error* route will be taken).

Configure HTTP Request

Details Headers Auth Response

Label

Method URL

Content-Type Charset

May be overridden by Content-Type on the Headers tab. Defaults to **application/json** if not set. Default is **utf-8**.

Body

Timeout (milliseconds)

If reached, the **Connection Error** route will be taken.

Cancel **Ok**

- Configure the HTTP headers to be sent with the request, one header per line.

Configure HTTP Request

Details Headers Auth Response

HTTP headers to be sent with the request, one header per line. Example:
Content-Type: text/plain; charset=utf-8
Accept: application/json
Setting Content-Length is unnecessary; it will be automatically computed. Content-Type only needs to be set if not specified on the Details tab.

Headers

Cancel Ok

- Configure if authorization is needed (e.g. Username/Password).

Configure HTTP Request

Details Headers Auth Response

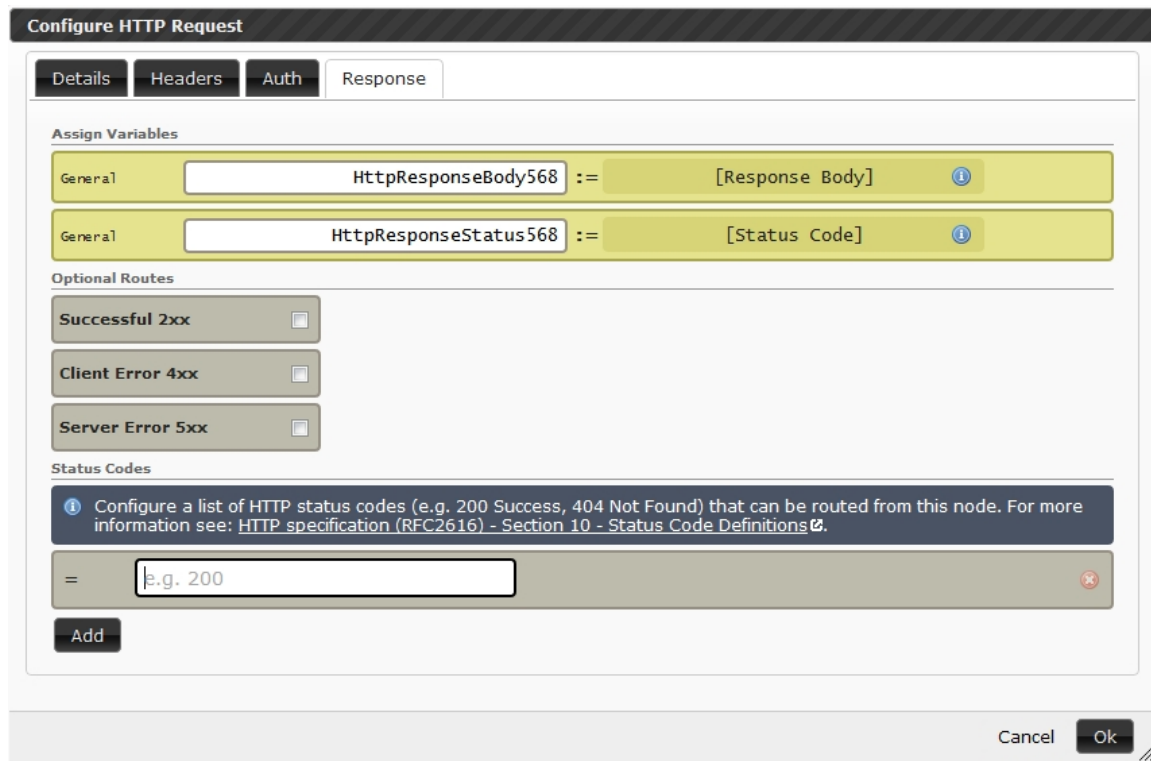
Type

Username

Password

Cancel Ok

- Configure what to do with the response: assign Variables, available optional routes (Successful 2xx, Client Error 4xx, Server Error 5xx) and status codes (e.g. 200 Success, 404 Not Found).



Configure HTTP Request

Details Headers Auth Response

Assign Variables

General := ⓘ

General := ⓘ

Optional Routes

Successful 2xx ☐

Client Error 4xx ☐

Server Error 5xx ☐

Status Codes

ⓘ Configure a list of HTTP status codes (e.g. 200 Success, 404 Not Found) that can be routed from this node. For more information see: [HTTP specification \(RFC2616\) - Section 10 - Status Code Definitions](#).

= ✖

Add

Cancel Ok

- A 'Default' route is also available.

4.28 Send Event

The Send Event node allows a user to configure the arguments to be sent to the Receive Event node.



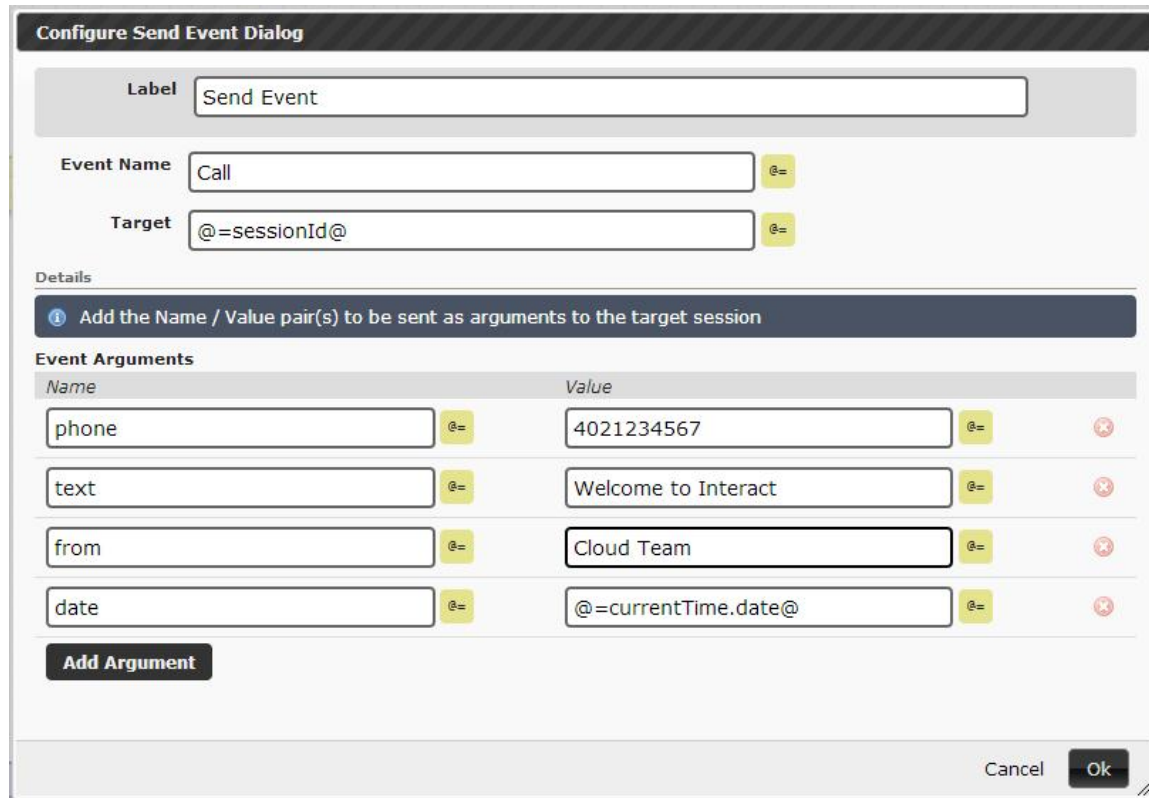
Send Event

⊞ [Empty]

[Empty]

Success ▶

Target Not Found ▶



Configure Send Event Dialog

Label:

Event Name: @=

Target: @=

Details

Add the Name / Value pair(s) to be sent as arguments to the target session

Name		Value		
<input type="text" value="phone"/>	@=	<input type="text" value="4021234567"/>	@=	✕
<input type="text" value="text"/>	@=	<input type="text" value="Welcome to Interact"/>	@=	✕
<input type="text" value="from"/>	@=	<input type="text" value="Cloud Team"/>	@=	✕
<input type="text" value="date"/>	@=	<input type="text" value="@=currentTime.date@"/>	@=	✕

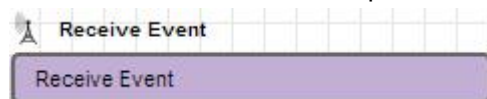
Add Argument

Cancel **Ok**

- **Event Name** - The name of the event. This is used in conjunction with the Receive Event node and as part of a variable in other nodes. *See example under Receive Event.*
- **Target** - The session id to send an event to. Using @=sessionId@ will get the value of the current session id.
- **Event Arguments** - The name/value pairs to send to the Receive Event node.

4.29 Receive Event

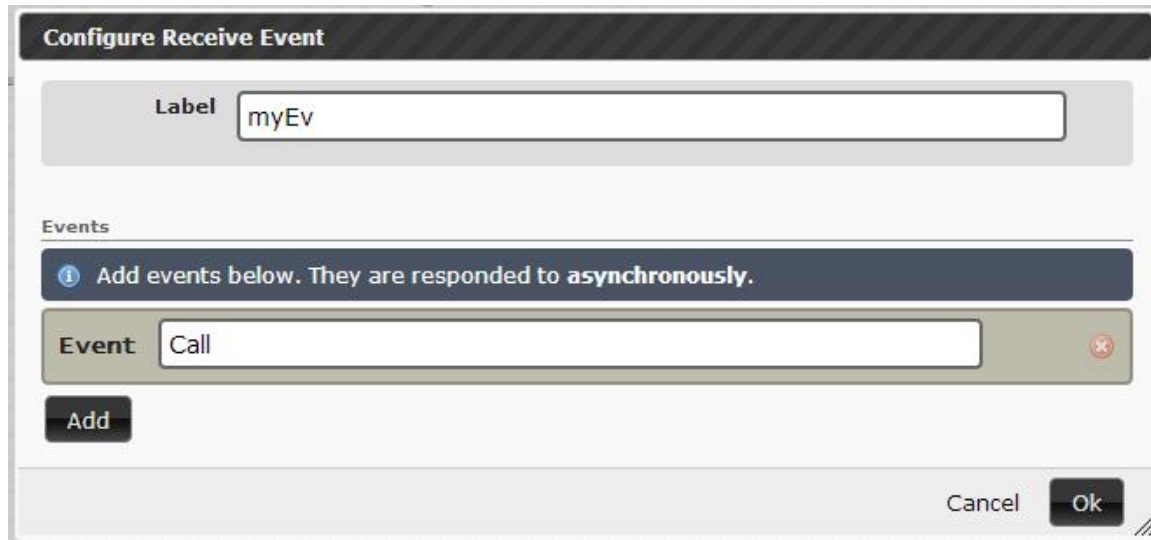
The Receive Event node receives the arguments configured in the corresponding Send Event node. This node is not available to subroutines. Instead, one is advised to receive the event in the parent application, and relay and re-enter the subroutine to process the event.



Receive Event

Receive Event

- Multiple Events can be configured on one Receive Event node.



Configure Receive Event

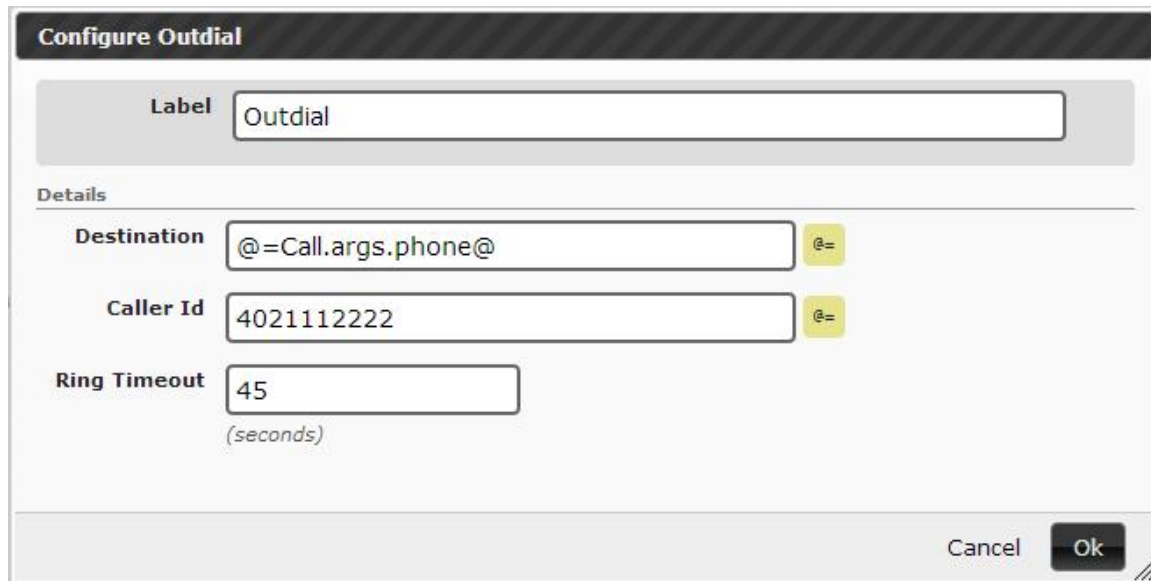
Label:

Events

*Add events below. They are responded to **asynchronously**.*

Event:

- **Event** - The name of the event. This must correspond to the Event Name configured in the Send Event node. All arguments received from the corresponding Send Event node are rolled up into an object call 'args'. This is then used as part of a variable in other nodes. The variable format is @=(Send Event name).args.(Event Argument name)@. See the example configuration for the Destination field of the Outdial node below:



Configure Outdial

Label:

Details

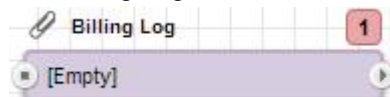
Destination: @=

Caller Id: @=

Ring Timeout:
(seconds)

4.30 Billing Log

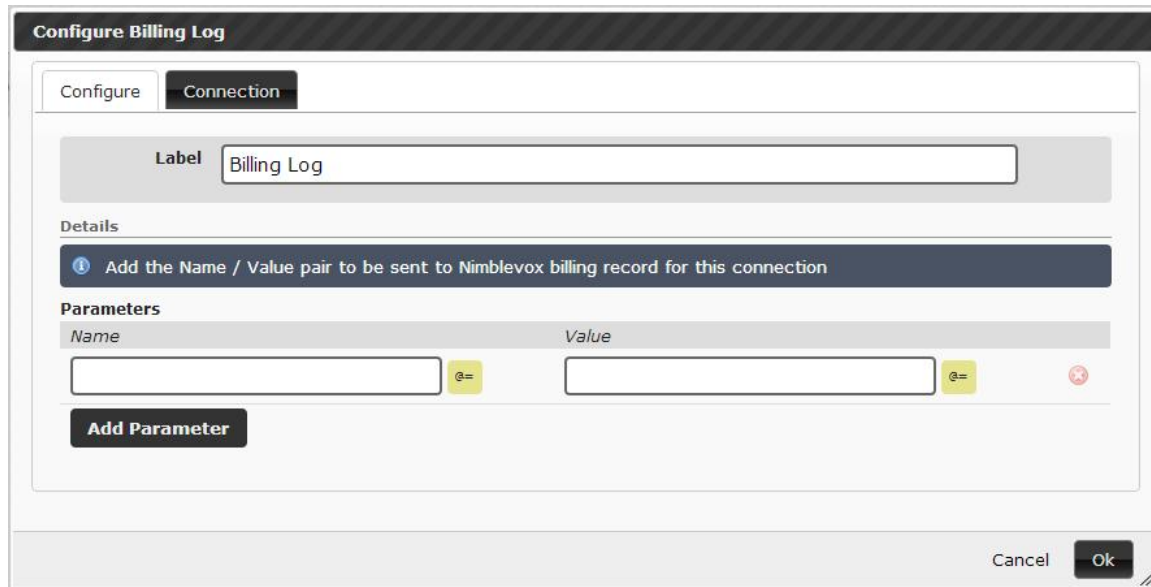
The Billing Log node allows a user to add additional field value to the billing record (*does not affect call flow*).



Billing Log

[Empty]

- The name and value can be literal or variable.



Configure Billing Log

Configure Connection

Label: Billing Log

Details

ⓘ Add the Name / Value pair to be sent to Nimblevox billing record for this connection

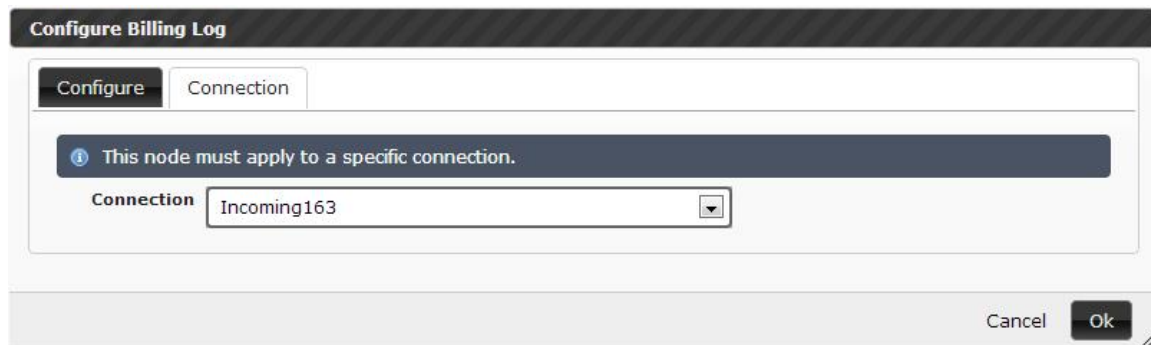
Parameters

Name		Value
<input type="text"/>	@=	<input type="text"/>

Add Parameter

Cancel Ok

- Must apply to a specific connection.



Configure Billing Log

Configure Connection

ⓘ This node must apply to a specific connection.

Connection: Incoming163

Cancel Ok

4.31 Comment

The Comment node allows a user to annotate something with a comment (*does not affect call flow*).



My comment...

- The comment displays on the call flow to allow you to give additional information.

4.32 Go To

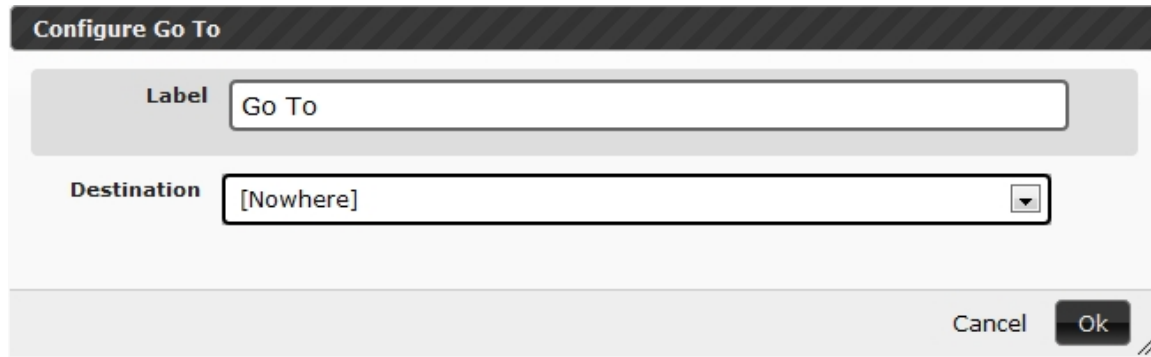
The Go To node redirects the call flow to a Target node.



Go To

<Not Set>

- This can be used to organize application call flows into multiple sections or pages.



Configure Go To

Label:

Destination:

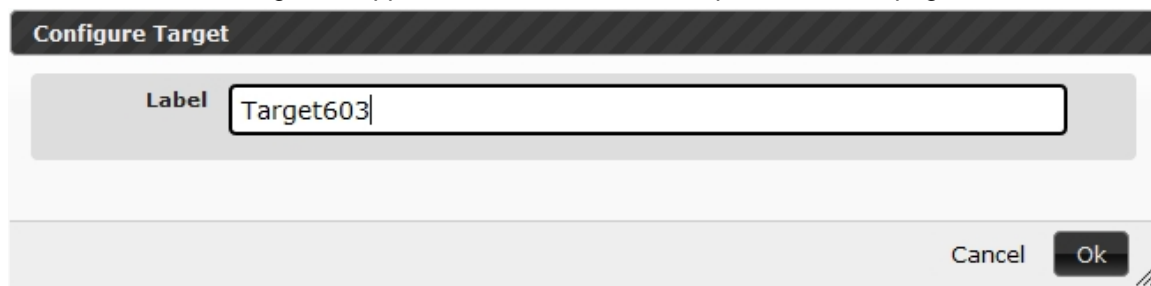
Cancel **Ok**

4.33 Target

The Target node is the named destination for a Go To node.



- This can be used to organize application call flows into multiple sections or pages.



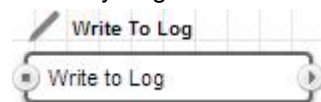
Configure Target

Label:

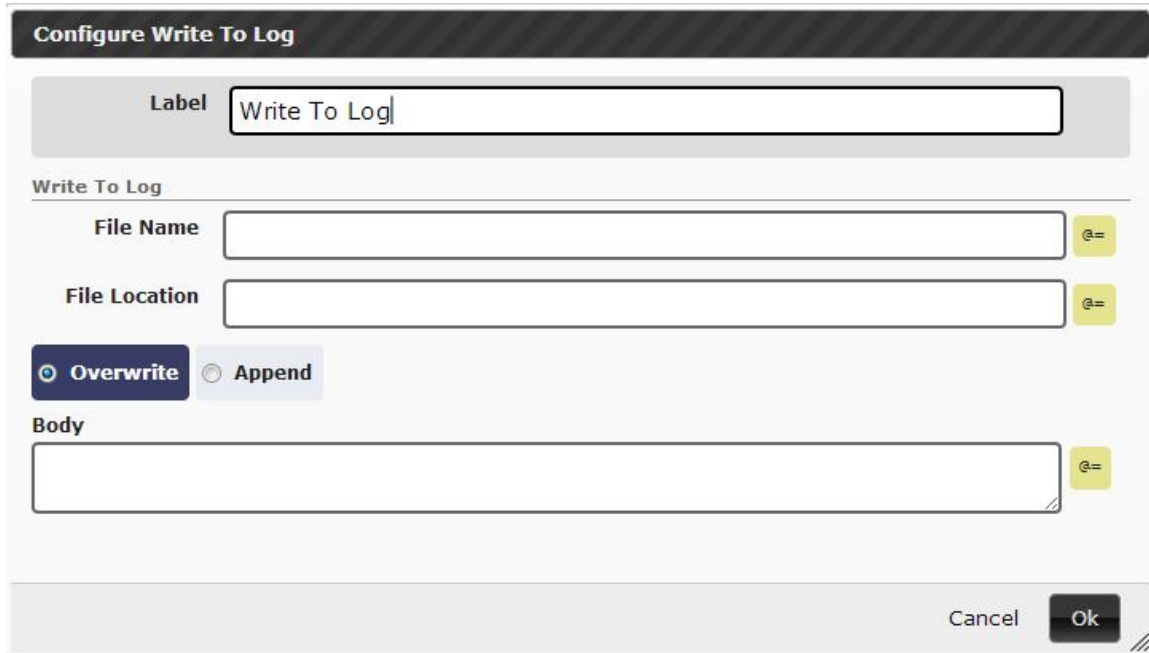
Cancel **Ok**

4.34 Write to Log

The Write to Log node is used to overwrite or append a log file. The file is written to the user's Build shared directory "logs".



- Configure the file name, file location and Body. The values can be literal or variables.




The dialog box is titled "Configure Write To Log". It contains a "Label" field with the text "Write To Log". Below this is a section titled "Write To Log" which includes a "File Name" field, a "File Location" field, and two radio buttons: "Overwrite" (selected) and "Append". There is also a "Body" field. Each of the three fields (File Name, File Location, and Body) has a small yellow icon with an equals sign to its right. At the bottom right of the dialog are "Cancel" and "Ok" buttons.

4.35 Fork

The Fork node is used to fork a single path into multiple paths, each of which will be executed asynchronously.



- To add additional forks, click the  icon.

4.36 End

The End node terminates the current application session.



5 Create Application


5.1 Add Node

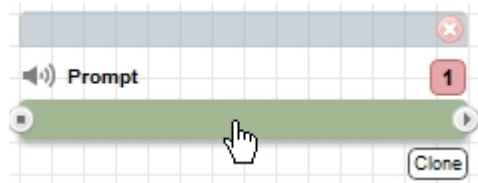
To add a new node, simply click anywhere on the canvas and all available nodes are displayed.



Click the desired node and it is added to the canvas.

5.2 Configure Node

To configure a node, move the mouse over the node until the  cursor is displayed.



Click and the node information is displayed.

Configure Prompt

Prompt Collect Connection

Label Prompt

Interruptable?

☒ Yes ☐ No

Prompts


+ Add Text-to-speech Audio file Audio URL Recording System audio

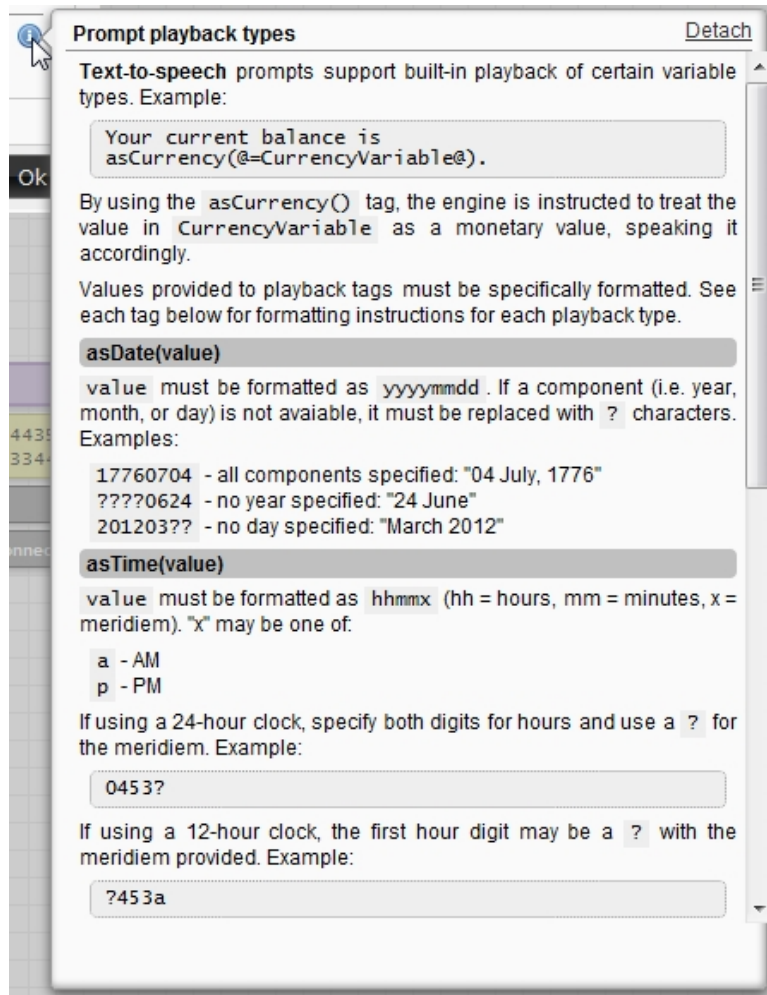
Cancel Ok

The fields and tabs displayed will vary for each node. (See descriptions of each node above)

Note: For some nodes, the Label will default to the name of the node. This can be changed.

5.2.1 Notes

All available notes for the node can be displayed by hovering over the  icon.



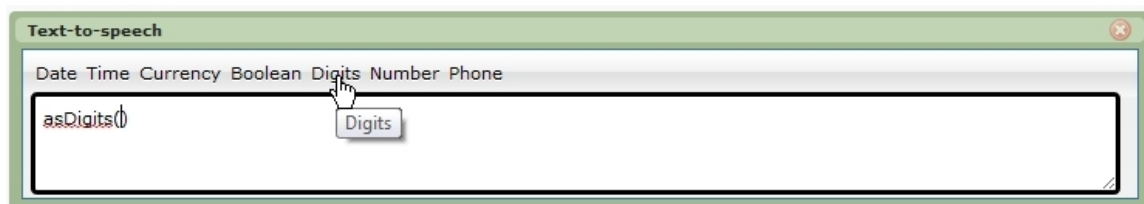
These notes can also be detached so that they can be viewed while editing the node.

5.2.2 Syntax Help

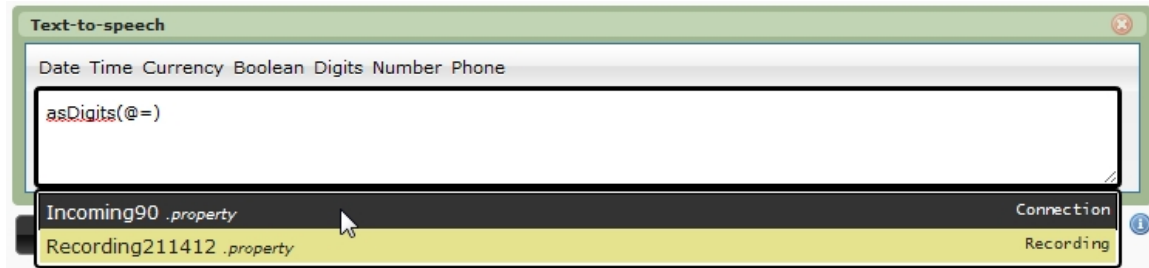
Where possible, we try to help out by providing the proper syntax for variables.

Example:

- When adding Text-to-speech, you can select one of the available variable types and the correct tag is added to the input field.



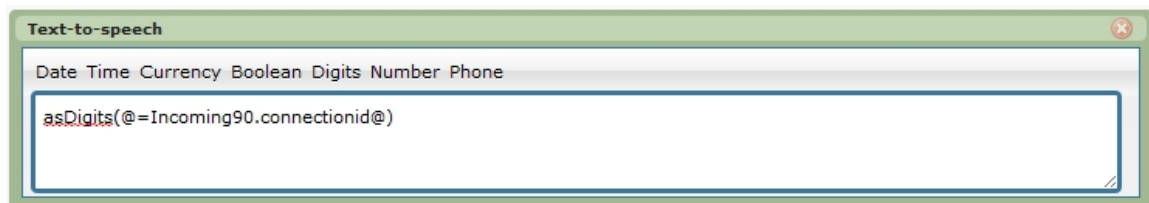
If using a variable for the value, start by entering '@=' . This will display all possible values.



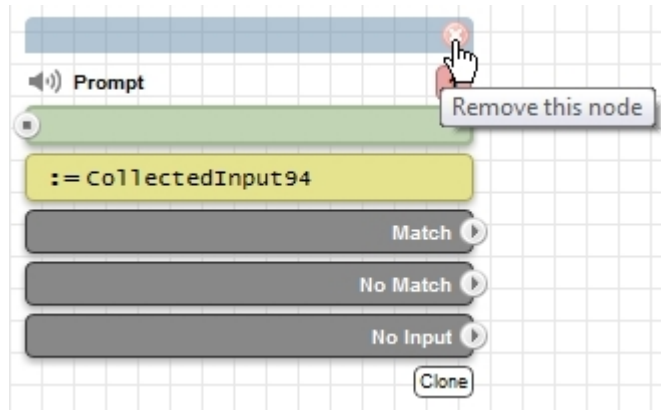
Click on the desired value. If additional options are available, they will then be displayed.



The syntax is then completed with a '@'.



- The



Click the  button and the node is deleted.


Notes:

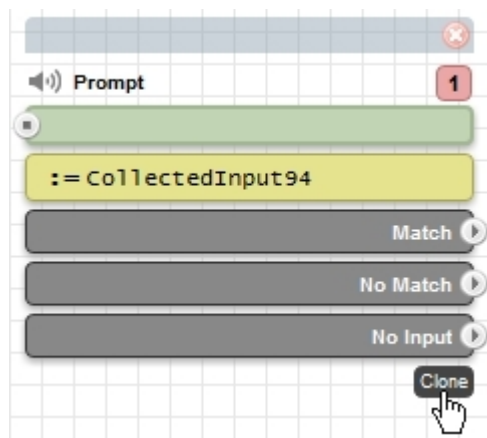
A node cannot be removed if its connection is referenced by another node.


The referencing node must first be deleted or modified to reference another connection.

A warning message is displayed which includes a link to the referencing node.

5.4 Clone Node

To create a clone of a node, move the mouse over the node until the  button is displayed in the lower right hand corner.

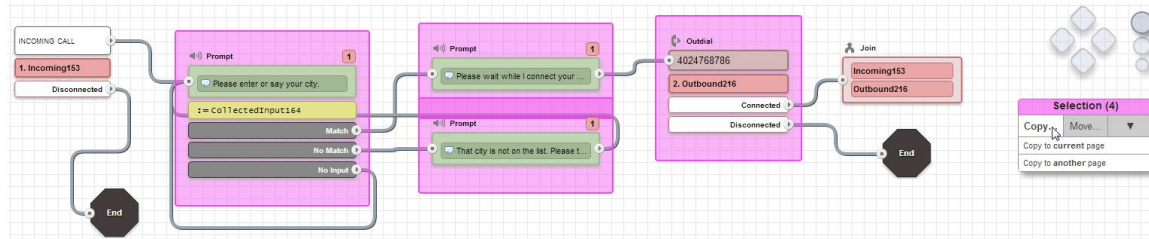


Click the  button and a clone of the node is created and placed on the canvas.

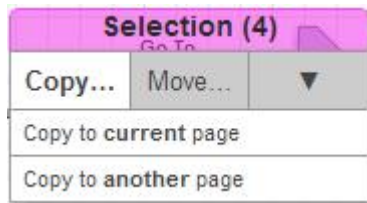
The new node can be edited/reconfigured and moved to the appropriate place within the application.

5.5 Copy/Move/Extract Multiple nodes

To copy/move/extract multiple nodes within the project, hold <Shift> key and select the desired nodes. A box with the available options is displayed.



Nodes can be copied to the current page or another page.



If another page is selected, pages names are displayed.



Nodes can also be moved or extracted to the current or another page. If another page is selected, pages names are displayed.

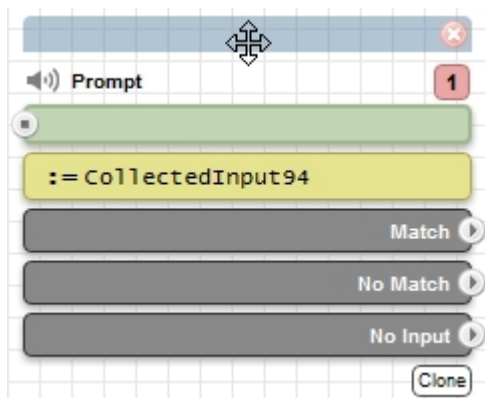


Move will simply move the selected area, where as, extract will move the selected area but retain functionality of any edges that connected into/out of the selected area by adding gotos/targets (*see example below*).




5.6 Move Node

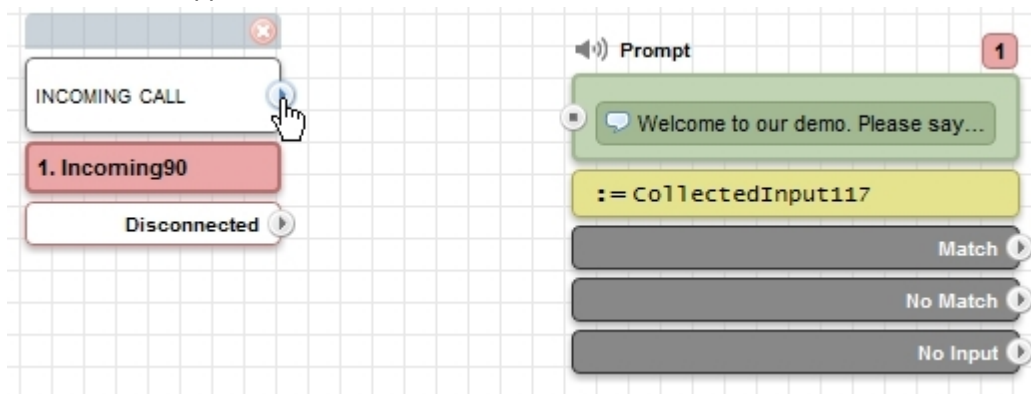
To move a node on the canvas, move the mouse over the node until the move cursor is displayed.



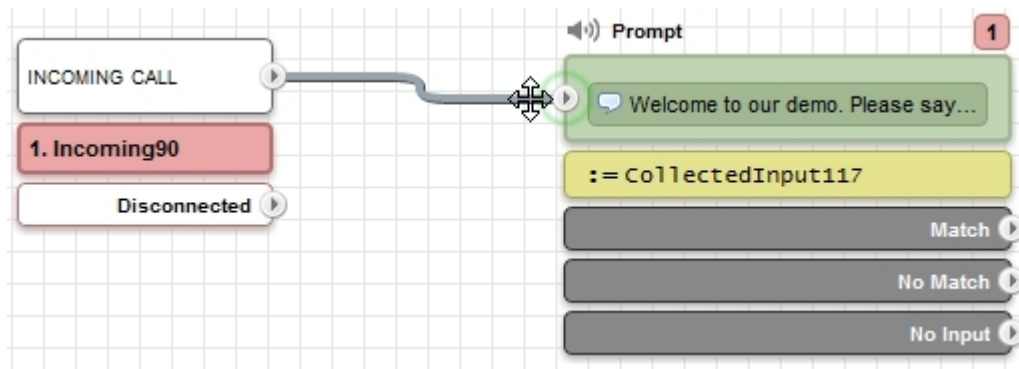
While holding the left mouse button, drag the node to the appropriate place within the application.

5.7 Connect Nodes

To connect two nodes, move the mouse over the outgoing arrow of the event you want to connect from until the  cursor appears.



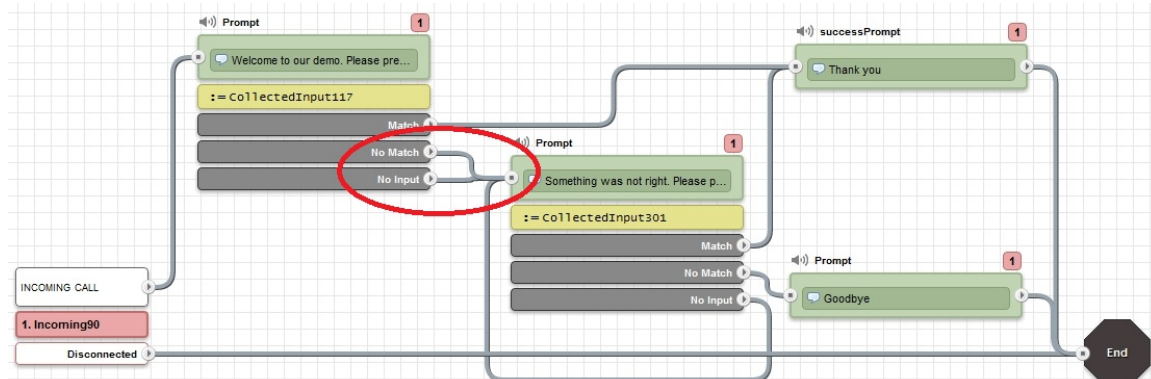
Hold down the left button and move the mouse to the node you want to connect to until a green circle surrounds the in arrow.



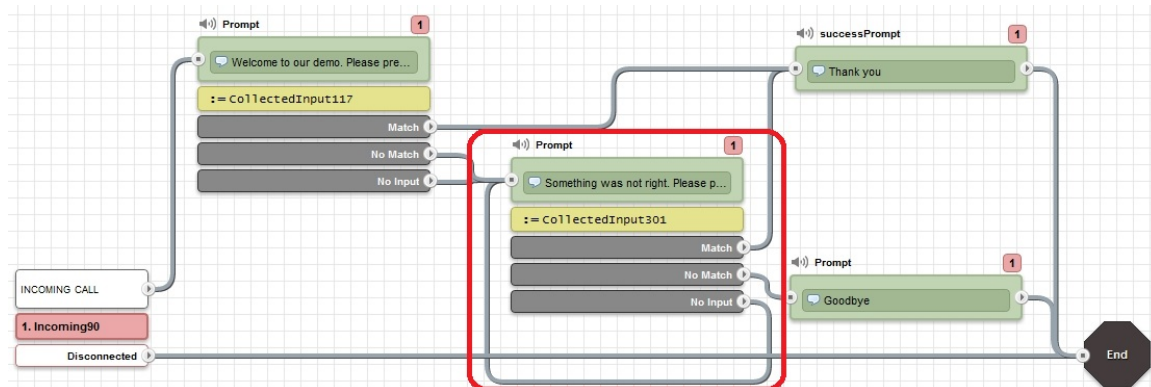
If you determine that this connection is not correct, move the mouse back over the outgoing arrow until the cursor again appears, hold down the left button and move the mouse to the correct node.

Notes:

- There can be a many to one relationship with the connections.
(e.g. The 'No Match' and 'No Input' routes can both connect to the same node.)



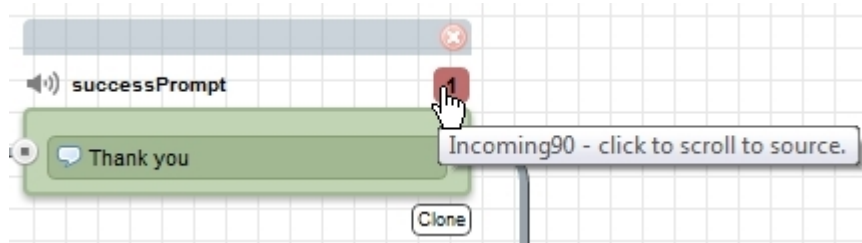
- You can also loop the connection back to itself.
(e.g. The 'No Input' route can connect back to play the prompt again until input is received..)



5.8 Return to Connection Source

All nodes that are required to apply to a specific connection have the ability to link back to that source.

Move the mouse over the **1** button until the cursor appears and click.



The screen jumps back to the source node, which is also highlighted in yellow.



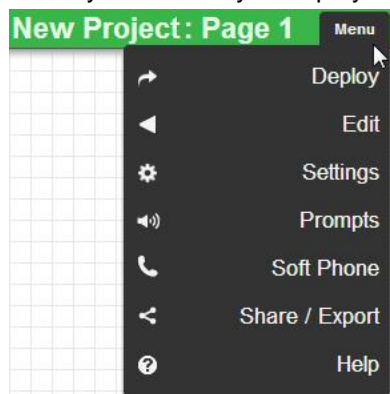
This can be handy for larger call flows that scroll off the page or are multiple pages.

6 Menu Button

6.1 Deploy

6.1.1 Deploy Application

When you are ready to deploy an application call flow for use, click the **Menu** button and then select Deploy.




The Deploy Application screen is displayed.

- With integrated Build systems, the Numbers listed are from your associated Nimblevox account (*there is a cost associated with numbers*) and Services are manually entered. Services are started by either an API call or web application (*there is no cost associated with services*).

 A screenshot of the 'Deploy Application' screen. At the top, there is a 'Production' checkbox. Below it is a section titled 'Numbers'. It contains a table with columns: 'Location', 'Application', 'Time', and 'Deploy'. There are two rows of data. The first row has '9981000200' in Location, 'Record Conference w/email' in Application, and '1/17/13 10:51:18' in Time. The second row has '9990003000' in Location, 'Custom grammars (imported)' in Application, and '1/17/13 10:39:16' in Time. Each row has a green 'Deploy' button. Below the table is a 'Services' section with a 'New service...' input field and a 'Deploy' button. Below that is a table with columns: 'Name', 'ID', 'Application', and 'Time'. There is a 'Close' button at the bottom right.

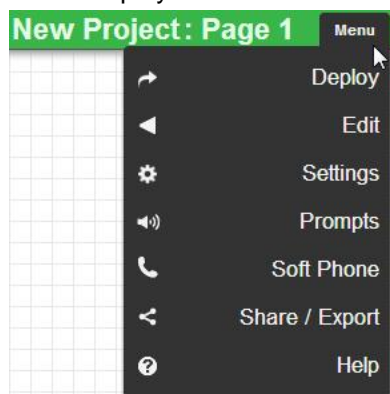
- The number(s) that still available will not have a group and application assign to.
- The number(s) that have been assigned to the group will have application assign to with the date and time of the last deployment.
- For Services, enter a description for the service and then click the **Deploy** button. The application is deployed to a service ID.

 A screenshot of the 'Services' section of the 'Deploy Application' screen. It shows a 'New service...' input field and a 'Deploy' button. Below is a table with columns: 'Name', 'ID', 'Application', 'Time', and a status icon. There is one row of data: '2' in Name, '531d16e2-b488-409f-ac24-07d65779a807' in ID, 'Campaign' in Application, and '11/30/12 16:25:11' in Time. The status icon is a green checkmark. There is a 'Close' button at the bottom right.


- Production checkbox is used with deploy when the application is production ready .  indicates the application is certified for production.

6.1.2 Remove Application from Number or Service

When you want to remove (*undeploy*) an application from a number or service, click the **Menu** button and then select Deploy.



The Deploy Application screen is displayed and shows the currently deployed applications.

- For Numbers, click the  button next to the application you wish to remove.

Location	Application	Time	
9985556666			
 Voice One 9984219988 9990003001	Campaign	11/30/12 16:24:21	
 CiteZone	Pay citation	11/30/12 15:22:04	 


A warning message is displayed.


Warning

Are you sure you want to remove the application from this number?

This will take immediate effect and cannot be undone.

Click the **Yes** button and the application is removed from the number.

- For Services, click the  button next to the name or the application you wish to remove.

New service...		Deploy	
Name	ID	Application	Time
2	 531d16e2b488-409f-ac24-07db5779a807	Campaign	11/30/12 16:25:11

A warning message is displayed.

Warning

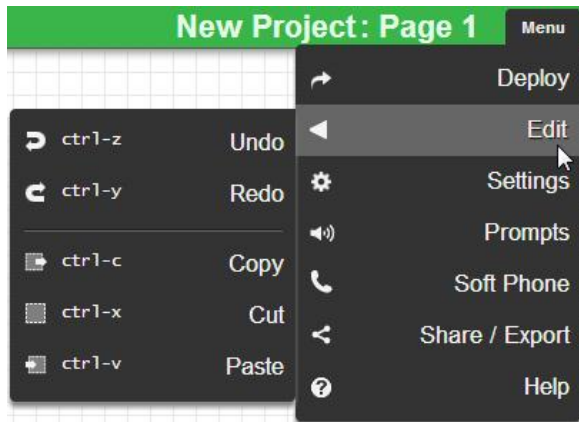
Are you sure you want to remove the application from this service?

This will take immediate effect and cannot be undone.

Click the **Yes** button and the application is removed from the service.

6.2 Edit

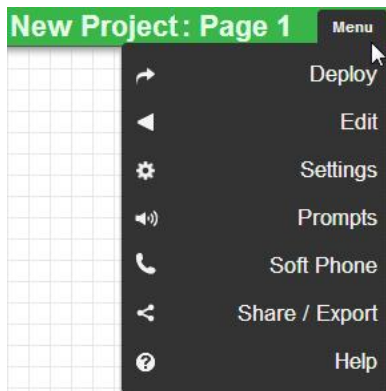
When you want to undo or redo the change click the **Menu** button and then select Edit.



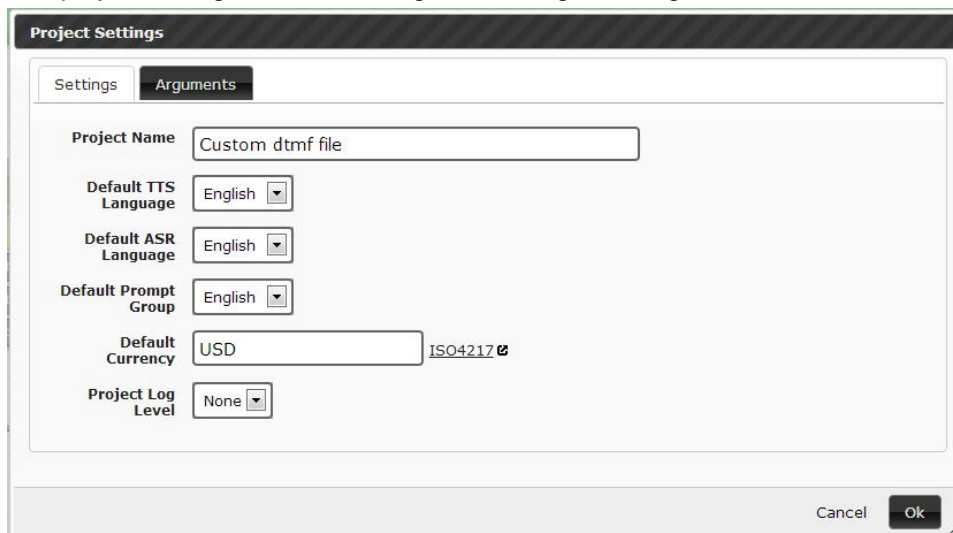
Click on the option. Short keys can also be used.

6.3 Settings

In the upper right hand corner, click the **Menu** button and then select Settings:



The project setting allows for changes to settings and arguments.



6.3.1 Settings

Settings consists of the Project Name, Default TTS Language, Default ASR Language, Default Prompt Group, Default Currency and Log level. Once the selections have been made they are saved by clicking the OK button.

6.3.1.1 Edit Project Name

The Project Name is under Menu/Settings. This name can be changed at any time.

6.3.1.2 Edit Default TTS Language

The Default TTS Language is under Menu/Settings. This is the default language used for TTS. The default language can be overwritten in the call flow application by setting <InboundConnection>.ttsLanguage in the Assign Variables Node.

6.3.1.3 Edit Default ASR Language

The default ASR Language is under Menu/Settings. This is the default language used for ASR. The default language can be overwritten in the call flow application by setting <InboundConnection>.asrLanguage in the Assign Variables Node.

6.3.1.4 Edit Default Prompt Group

The Default Prompt Group is under Menu/Settings. This is the default prompt group for playing audio files. The Default Prompt Group can be overwritten in the call flow application by setting <InboundConnection>.promptGroup in the Assign variables Node.

6.3.1.5 Edit Default Currency

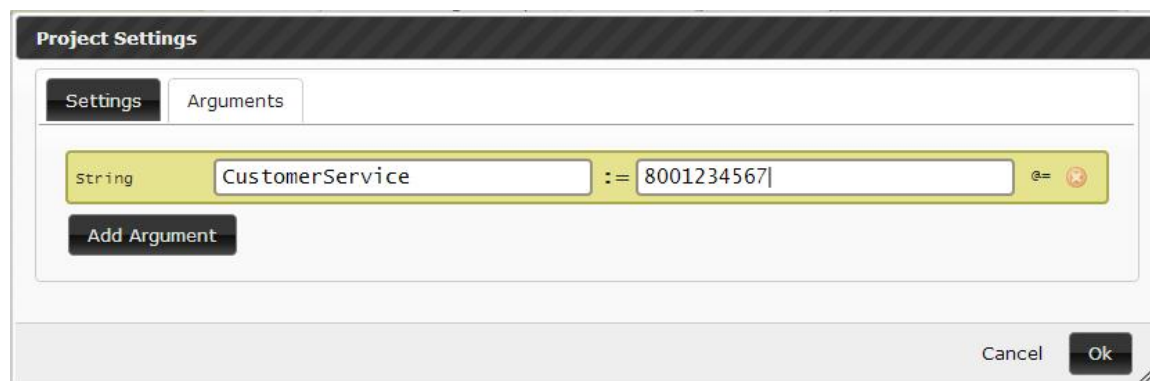
The Default Currency is under Menu/Settings. This is the default currency that is played back when asCurrency() is used when playing TTS. The Default Currency can be overwritten by passing currency type to asCurrency(). Example: asCurrency(1234, USD).

6.3.1.6 Edit Project Log Level

The default project log level is under Menu/Settings. The log is written to logs directory under Storage tab on Nimblevox.

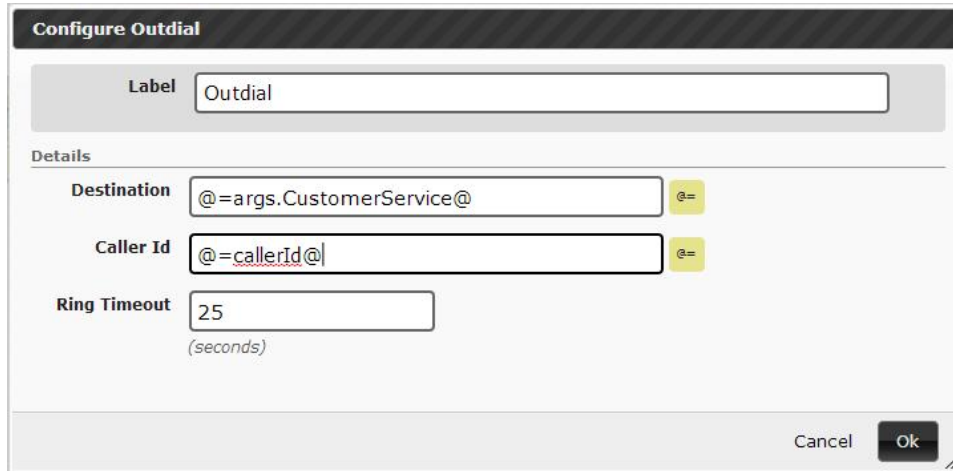
6.3.2 Arguments

The argument tab allows a user to create variable(s) to be used within the project. This is the default value and can be overwritten on the Nimblevox side.



The screenshot shows the 'Project Settings' dialog box with the 'Arguments' tab selected. The dialog has two tabs: 'Settings' and 'Arguments'. Below the tabs, there is a list of arguments. One argument is visible: 'String' with the variable name 'CustomerService' and the value '8001234567'. To the right of the value is a small icon with '@=' and a red 'x'. Below the list is a button labeled 'Add Argument'. At the bottom right of the dialog are 'Cancel' and 'Ok' buttons.

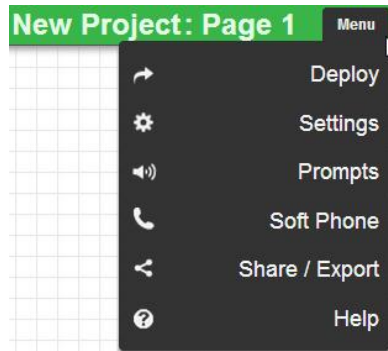
To access the argument variable the format is @=args.(variable)@. See example below.



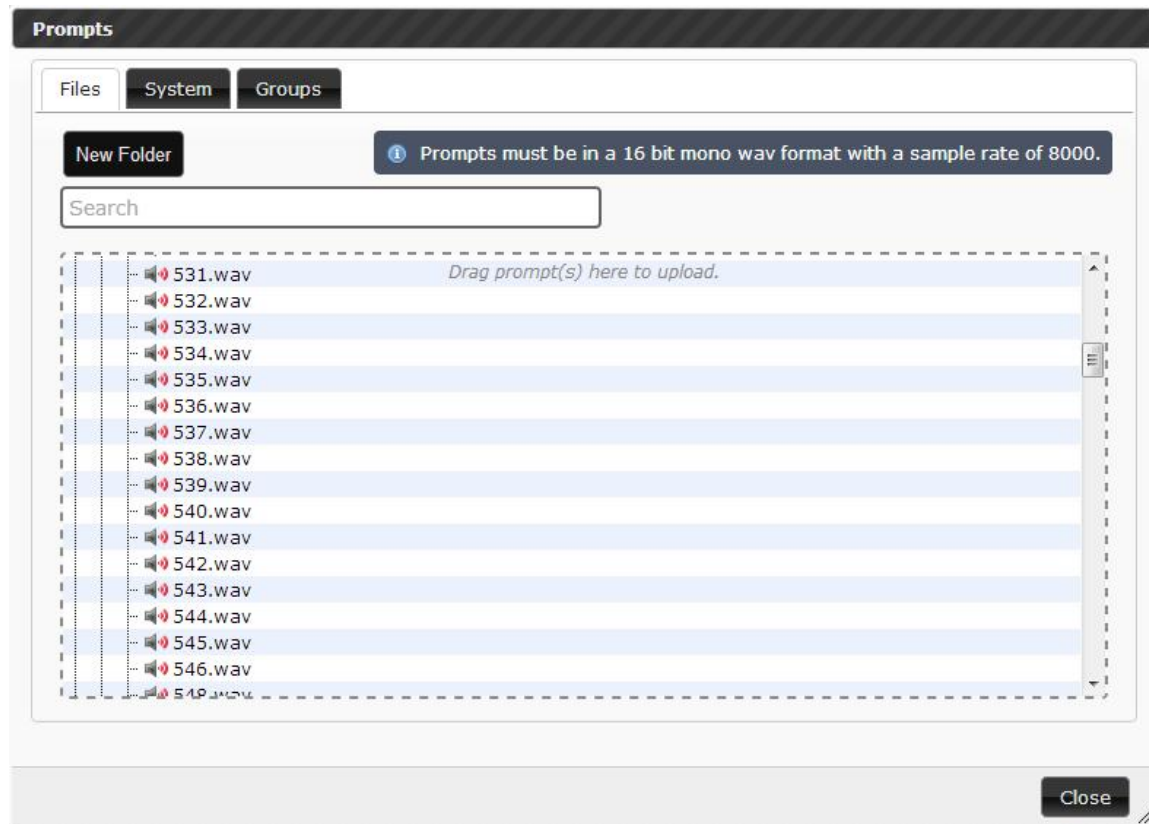
The 'Configure Outdial' dialog box is shown. It has a title bar 'Configure Outdial'. Below it is a 'Label' field with the value 'Outdial'. Under the 'Details' section, there are three fields: 'Destination' with the value '@=args.CustomerService@', 'Caller Id' with the value '@=callerId@', and 'Ring Timeout' with the value '25' and a note '(seconds)'. At the bottom right are 'Cancel' and 'Ok' buttons.

6.4 Prompts

In the upper right hand corner, click the **Menu** button and then select Prompts:



The prompts/files tab allows for prompts to be uploaded to Build by simply dragging the audio file onto the screen. Once the prompts are uploaded they can be used by any Build application. Uploaded prompts reside in `/interact/users/<userId>/public`.



The Prompts/System tab allows for system file audio to be uploaded to Build by simply dragging the audio file onto the screen. System file audio is concatenated together to form dates and numbers. Once system files are added to a prompt group, prompt groups can then be assigned to applications.

Prompts

Files System **Groups**

Drag files below to upload. Prompts must be in a 16 bit mono wav format with a sample rate of 8000.

Prompt group: English

prompt	file name		
zero	0.wav		Delete
one	1.wav		Delete
two	2.wav		Delete
three	3.wav		Delete
four	4.wav		Delete
five	5.wav		Delete
six	6.wav		Delete
seven	7.wav		Delete
eight	8.wav		Delete
nine	9.wav		Delete
ten	10.wav		Delete
eleven	11.wav		Delete

Close

The Prompts/Groups tab allows for a friendly name to be assigned to a Prompt Group Language.

Prompts

Files System **Groups**

New Group

Prompt Group Language: Spanish

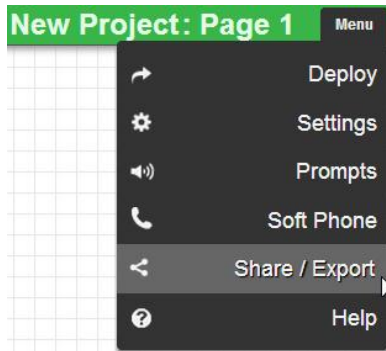
Current Prompt Groups

English(English)	(default)
Spanish(Spanish)	Delete

Close

6.5 Share/Export

Users can share project(s) from Menu/Share/Export.



The Download Project button allows a user to download the project that is currently displayed on the canvas. Once it is clicked the project will be downloaded to the users local drive.

The Create Link button creates a link the user can share with other users. When follow the link, a copy of the project will be imported to the account.



6.6 Soft Phone

Note: This feature is only supported for Chrome and Firefox. It is not available in IE.

A soft phone is available to test Build applications. The numbers that are assigned to a Nimblevox user are shown in the soft phone drop down menu. The soft phone will dial the number chosen and the application that is currently assigned to that number will be invoked.



From the soft phone it is also possible to Deploy the application that is currently on the canvas to the number that is chosen on the soft phone.



Deploy To Test allows user to deploy the application to the test number.

Download Log allows user to download the softphone log. Log level must be configured in Menu->Settings->Project Log Level to write to file.

7 Build Processes

There are essential processes for Build to function properly. They are:

- vipStart - service that starts and stops all the Nimblevox Engine processes
- cassandra - a database used to store Nimblevox Build and other application specific information
- vex - a server that works in conjunction with the Nimblevox Engine to process the application call flow (as *defined on the graph*) in real time
- phonebook - a server that stores and processes information about the different projects created through Nimblevox Build
- collector - receives statistics from all system applications that are configured to send status information.
- dmv - The Dynamic Machine Validation (DMV) provides the license management for each host system running product software. It connects to the Web Licensing Server on an hourly basis to verify the license validity.
- httpd - Apache Hypertext Transfer Protocol Server used by the Nimblevox Engine to manage internal and external web request.

Each process can be started, stopped, restarted or monitored using the service script.

Starting: service <process name> start

ex. service vipStart start

Stopping: service <process name> stop

Restarting: service <process name> restart

Monitoring: service <process name> status

Note: When restarting, services start in the order listed above. If the system needs to be renamed be sure to stop the dmv process first.



NewNet Communication Technologies, LLC
2 Trap Falls Road, Suite 504, Shelton, CT 06484
Tel. +1 203.926.0049 Fax. +1 203.647.0580

Chicago
NewNet Communication Technologies
Arlington Ridge Office Center
3223 N. Wilke Road
Arlington Heights, IL 60004
+1 847.253.9200

Lincoln
NewNet Communication Technologies
Centre Terrace
1225 L Street, Ste. 300
Lincoln, NE 68508
+1 402.476.8786

India
NewNet Techno Engineering India Pvt Ltd
G001 Ground floor, Sigma Soft Tech Park
Ramagondanhali Varthur Hoblid
Bangalore 560066
+91 80 41528312

Romania
NewNet Communication Technologies
str. Dr. Iacob Felix nr. 70
Sector 1
Bucharest
Romania