IVR Studio User Guide

Purpose:
This document describes how to use the IVR Studio visual editor to edit and maintain call flows.

Table of Contents

Revision History

Revision	**Date**	**Description**	**Author**
1.0	10/29/14	Initial Release	J. Don

1.0 Introduction

This document explains everything you need to know to build IVRs with IVR Studio. The document is broken down in three sections based on the user's level of familiarity with IVR Studio.

1.1 Getting Started

A no-nonsense guide to the basics of working with IVR Studio. A good starting point for those new to the CCI platform or those who don't want to miss a thing. Those familiar with CCI may want to skip this section and just dive right in.

1.2 Node Reference

Detailed descriptions of each node and their fields. A good place to start for those who want to get right to building with the IVR Studio editor and just check back here when you come across a field you don't recognize.

1.3 Scripting

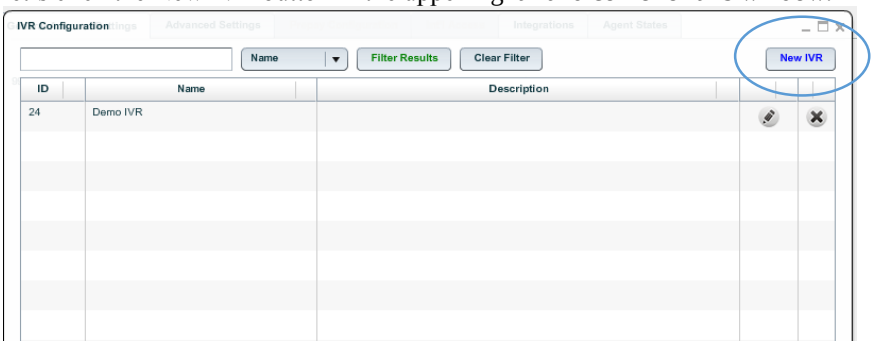
Use JavaScript to take your IVRs to the next level. Includes general information on how to use variables throughout your IVR, examples of common tasks such as parsing web service results, and documentation of the `ivr` API scope.

2.0 Getting Started

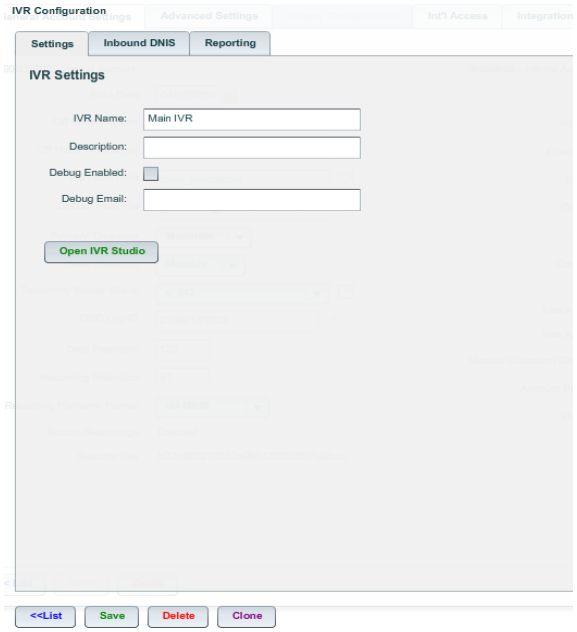
To get started, log into the CCI Management interface. Detailed information on the use of the management interface is described in the CCI Admin Users Guide and is beyond the scope of this guide.

- Log into the account you wish to create an IVR for
- Navigate to Configure > IVR Studio

A new window will pop up on screen. This window is the IVR Configuration window, and will contain any already in-progress/completed IVRs created in the IVR Studio. Since we are just getting started, let's click the New IVR button in the upper-right-hand corner of this window.



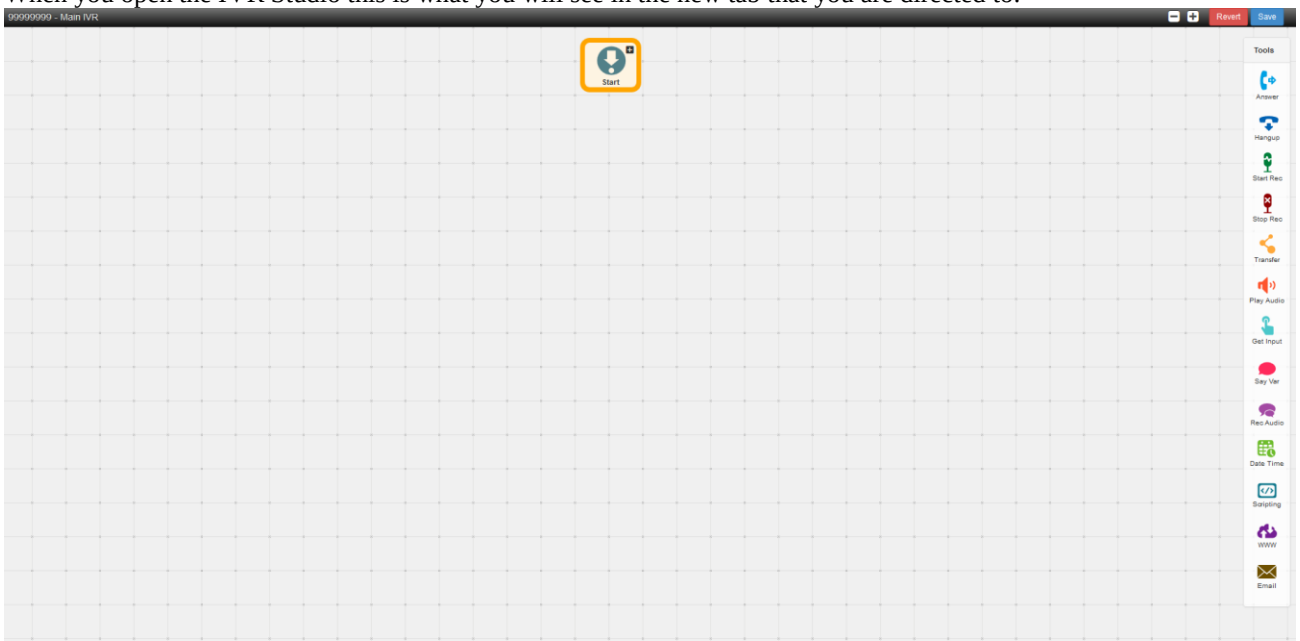
Clicking this button will change the window to an IVR configuration window where you will be required to assign your IVR a name. Once you have given your IVR a name, click Save.



Once you have clicked save, after giving your IVR a name, you can assign the IVR's Inbound DNIS or assign multiple DNISs by clicking the Inbound DNIS tab. Or if you want to get straight to building your IVR you can simply click on the Open IVR Studio button.
When you open the IVR Studio you will automatically be directed to a new tab in your browser where you will be able to visually plan out and build your IVR to suit your needs.

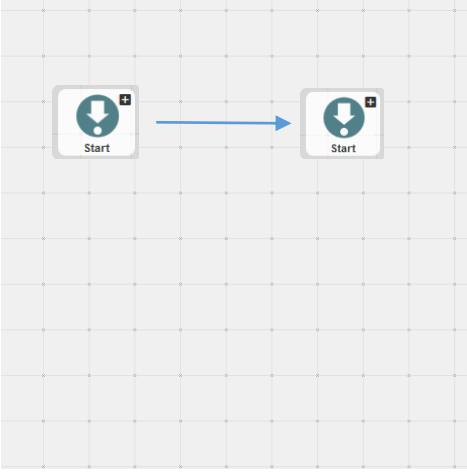
Inside the IVR Studio

When you open the IVR Studio this is what you will see in the new tab that you are directed to:



This is where you will design, edit, and build your IVR to meet your custom requirements.
From here we will walk through setting up an IVR using the IVR Studio environment. This walkthrough will showcase most, though not all of the features built in to the IVR Studio. There are some features that are not often used and are for very specific set-ups and situations, and as such will not be a part of this general walkthrough. For more information on all of the features available in the IVR Studio, please see the Features section of this guide.
After going through this walk-though you should have a good foundation that you can use to build your own IVR based around your specific requirements. With that in mind, let's take a look at the key components and mechanics of the IVR Studio environment in which we will be working.

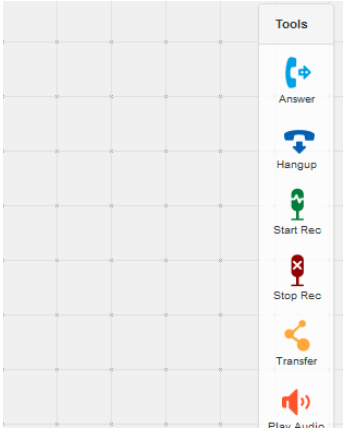
The Studio Environment

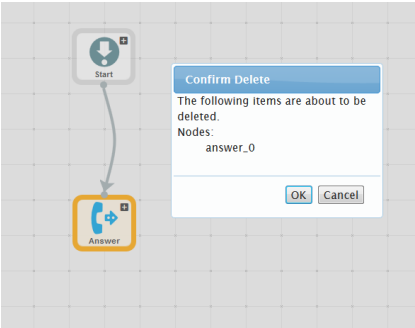


First off, we have **the grid**, or the background upon which all of our tools, and our overall setup will sit.

- Anything that is on the grid (**nodes** and **tool pallet**) can be moved to any other place on the grid by simply clicking and dragging.
- The grid itself, or your view of the grid can be moved in the same way (click anywhere on the grid itself and drag until you have the piece you need to work with, in view)

Next we have the tool palate. This is where you will find all of the tools needed to create your functional IVR. To add a tool to the grid to be incorporated in your IVR, click on the icon and drag it to any space on the grid and a node will appear on the grid.
Any instance of the tools that has been added to the grid for use, is referred to as a node and will appear inside its own clickable and moveable box on the grid. There can be multiple nodes of the same type that can be interconnected and manipulated to work with one another. The Start icon that you see when you start building any IVR is a node that cannot be created via the tool palate, nor deleted from the IVR environment. It is required for any IVR to function and as such is unique in this way.

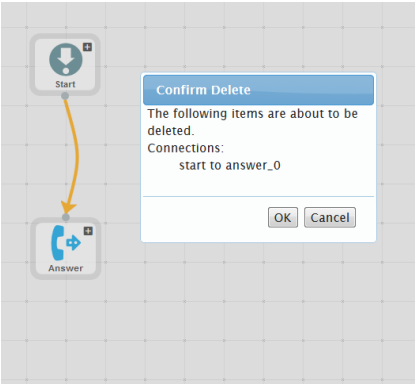




- In order to delete or remove a node from the grid or your setup, select the node and hit the "delete" key (Note: the "backspace" key will have no effect, it must be the "delete" key). This will be the same for connections between nodes (more on these shortly).

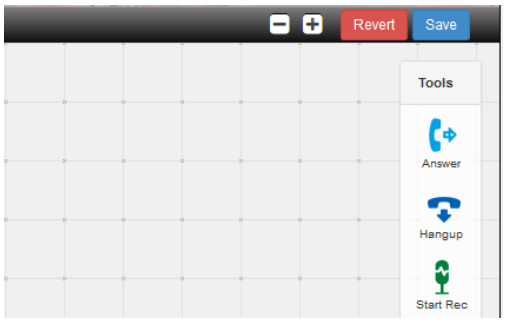
- Make sure that the component you are removing is the one you intend to; hitting the delete key will remove the currently selected component.

- The component that is currently selected will be highlighted in yellow-orange



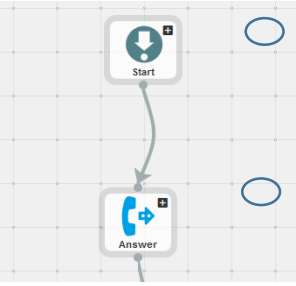
- The IVR Studio will prompt you to confirm your deletion

Next we have the four buttons in the upper right-hand corner of the window. The "-" and "+" buttons will zoom out and in, respectively, on the grid and it's components. This can also be done via the scroll wheel on your mouse.
The "Save" button will save your current setup, while the "Revert" button will allow you to go back to your last saved setup. Any changes that are made between the last time "Save" was clicked and "Revert" is clicked, will be removed. If the setup has not yet been saved, "Revert" will put everything back to default.

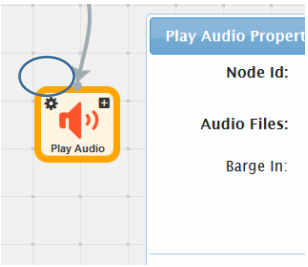


Nodes and Connections

Some nodes will have only one function, while others will be more complex. For instance, the Answer node will have only one function: to answer the call. Whereas the Play Audio node can be configured with specific properties that apply to your situation. For more on each node's specific set of functions please see the Features section of this guide.



For nodes with only one function, there will be only one button, the "+" button in the upper right-hand corner of the node icon. This button will be present on all nodes and is what will allow you to connect that node to any other node. Note: some nodes can only have one connection (Start, Answer, Hangup), whereas others can have as many connections as you can come up with.



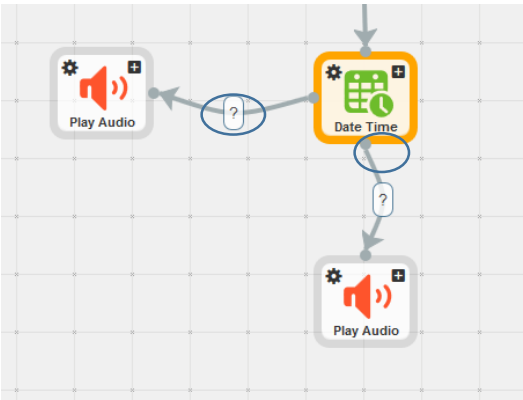
Any node that has specific properties that you will apply for your situation, will have a gear button in the upper left-hand corner of the node icon. In order to access and change the properties for that node, double-click this button and the properties window for that node will pop up on screen.

Connecting nodes will directly influence the flow of your IVR. There are a few things to note about connecting nodes and various functions that connections will have. Connections between certain nodes will yield new properties based on the interactions between the various nodes. A little on that next, but first let's just connect two nodes.

To connect a node:

- Click the "+" button and drag off of the icon until an arrow appears
- Continue to drag this arrow until you reach the node you wish to connect to
- Let go at when you have brought the arrow on top of the node you wish to connect to
- The direction of the arrow indicates the flow of the call through the steps in the IVR.

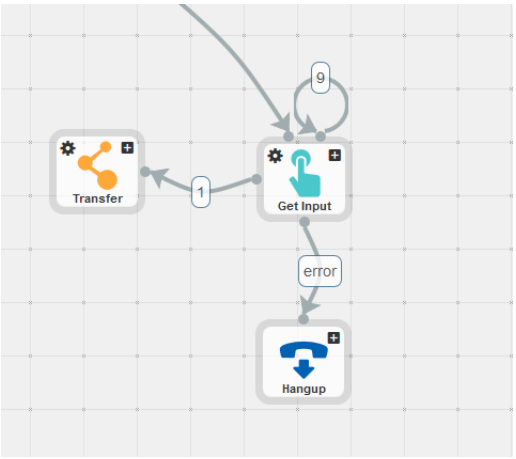
Getting back to a property of connections that was alluded to earlier: Some connections will have their own properties that will result from the interactions between the nodes that they connect. When this is the case, the connection will show a "?" in a small bubble in the middle of the connection. This is to indicate that there are options and settings that need to be manipulated in order for the connection to have meaning.



In order to access the properties for the individual connection you want to configure, double-click the bubble with the "?" and the properties window will pop up on screen.

Note: You can have the call loop back to the node it just came from by creating a loop. This is done by:

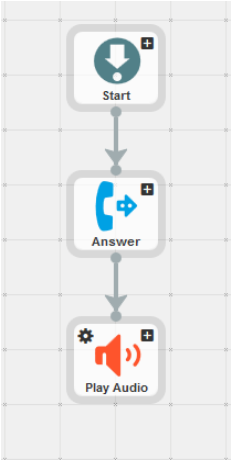
- Clicking the "+" button
- Dragging off of the node, then back to that same node and letting go.
- If done correctly, there should now be a connection that loops back to its original starting node. This can be useful for a DTMF menu where pressing "9," as set up in the figure to the right, brings the caller back the menu options for that node.



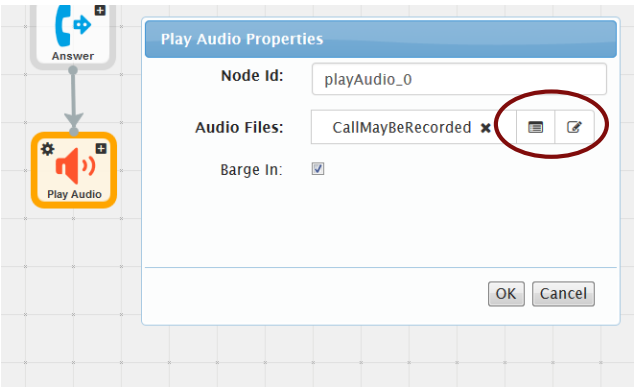
Building an IVR

Now that we have a solid understanding of the environment and the mechanics of the IVR Studio, we can begin constructing a fully functional IVR. As mentioned earlier, this will cover many, though not all of the features that are available in the IVR Studio. With that in mind, let's begin:

We will starting from scratch with our grid having only the Start node and tool palate in place



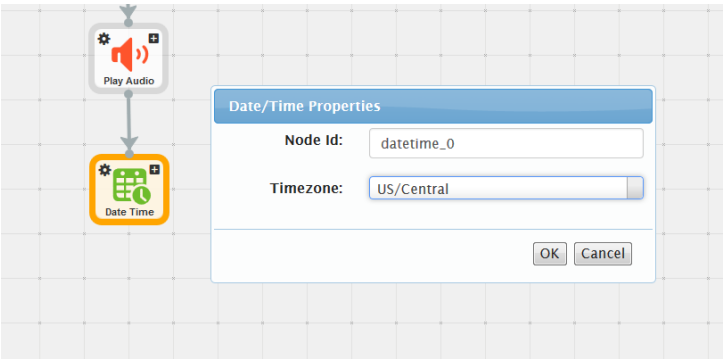
- Let's start off by adding an Answer node as the second step of the IVR. In general this is how most IVRs will be set up. **Note:** there are certain instances where one may want to have the IVR perform a certain action (send an HTML Post, run a Javscript) before picking up the call, those these are rare and usually very specific
- Connect the Start node to the Answer node
- Add a Play Audio node to the grid
- Connect the Answer node to the Play Audio node
- Double-click the gear icon to open the properties window for the Play Audio node



- Here we would assign an audio file that would play something to the effect of "Thank you for calling. Please not that your call may be recorded or monitored for quality assurance"
  - There are two buttons that allow you to either search the audio library, or manually add the file if you know the name of the file.
  - Hovering your mouse over the button will reveal what the button does.
  - You can assign more than one audio file to be played here if you so choose by repeating the process.

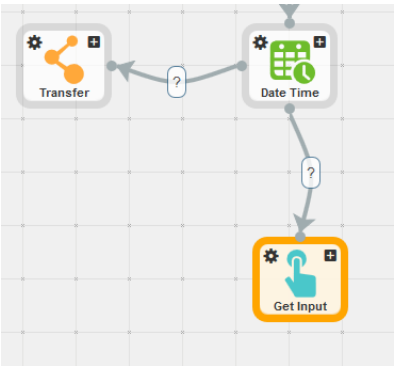
Next let's add a schedule to our IVR. For this example, we'll assume that we have a specific gate where all calls after-hours will be sent. In order to get this set up, we will need to do a few things:

- Add a Date Time node to the grid
- Connect the Play Audio node to the Date Time node
- Double-click the Date Time node
- If you like, to better keep track of nodes, you can give this node a specific name in the Node Id field
- Assign a Timezone to the node (for this example we will assign US/Central)
- Click OK

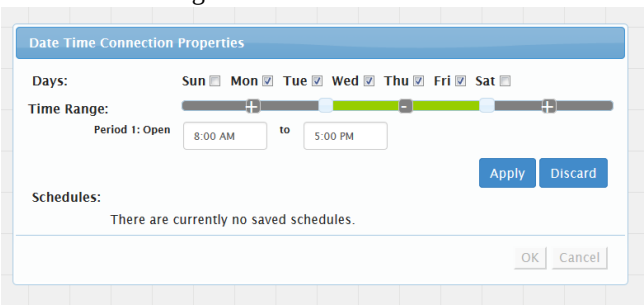


Now, depending on how many different schedules you have, and destinations that you want to send to based on those schedules, there are a few ways that we can proceed with the next step. For this walkthrough we will assume a basic set up where we have one main schedule, and an after-hours schedule. With this in mind we will set this up as follows:

- Add a Transfer node and a Get Input node to the grid
- Connect the Date Time node to the new nodes (one connection to each. Order in which you do this will not have any bearing on the routing)
- Notice both of the new connections have "?" bubbles. As described earlier, this means we now have to change the properties of these connections to allow the call to flow in the way that works for our goals
- In this case, we will double click the connection to the Get Input node



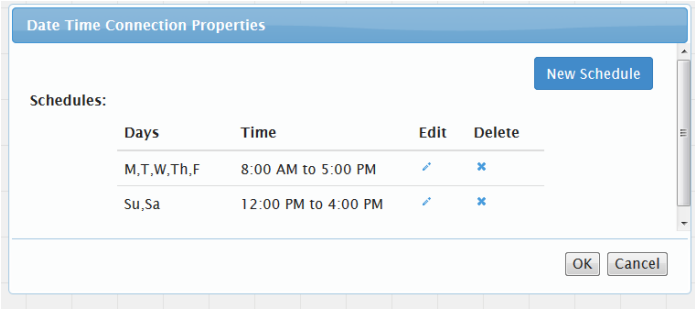
Double-clicking on this connection will yield a new window with a heading of Date Time Connection Properties. Notice that inside this window there is a message "Schedules: There are currently no save schedules," and two buttons: Out of Schedule, and New Schedule. Click the New Schedule button. Now the window will change to allow us to create a schedule for this connection.



- For this example, let's assume our regular call center is open Monday – Friday from 8 AM to 5 PM, and Saturday – Sunday from 12 PM to 4 PM.
- Check the boxes next to Mon, Tue, Wed, Thu, and Fri
- Click the "+" box on the line next to Time Range
- Now we can either use the sliders to either side of the "-" box to set our time range for Mon – Fri, or we can use the boxes below to enter the times manually
- Once the times are set, click the Apply box
- You now have a schedule for Mon-Fri

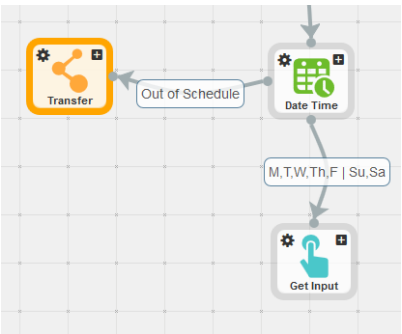
Next we will set the schedule for Sat and Sun

- Click New Schedule
- Repeat the steps above, only now for Sat and Sun for 12 PM to 4 PM
- You should now see a window that looks like the figure to the right
- Click OK to apply these settings to this connection



Now that we have our standard open schedule set, we can tell the IVR what to do when a call comes in outside of that timeframe.

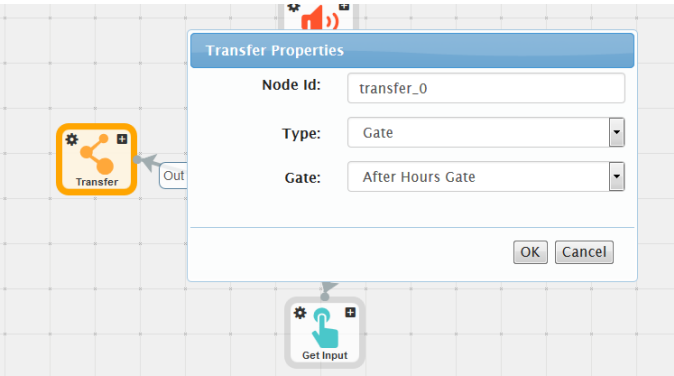
- Double-click the "?" bubble on the connection to the Transfer node
- Click the Out of Schedule button
- Click OK



By setting this connection to Out of Schedule, the Date Time node will check for any connections to that node with set schedules. If the call is coming in to the IVR at any time outside of any of the other schedules that are connected to this Date Time node, the call will default to the Out of Schedule path.

Next, let's take a look at the transfer node to tell the call where we would like it to go when it comes in after-hours.

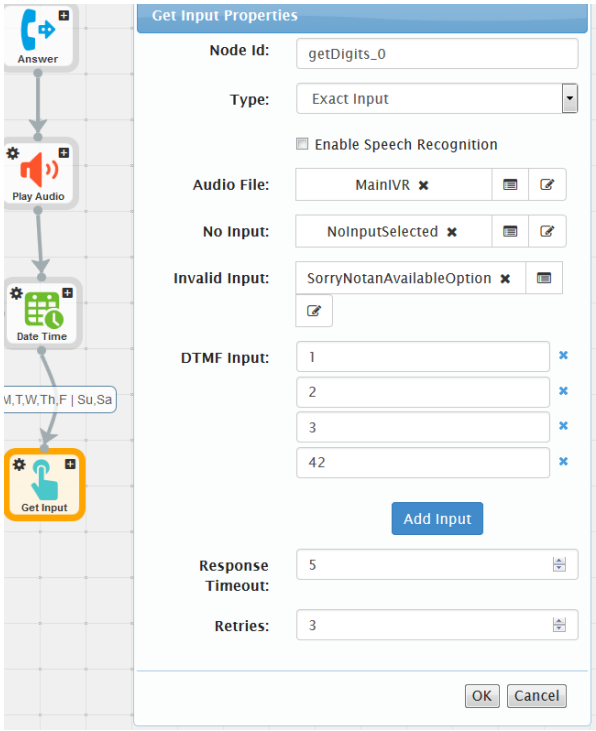
- Double-click the Transfer node
- We now have a choice of how we would like this call to be transferred by clicking the dropdown next to Type
- For this example, let's assume we have a gate set up to handle the after-hours calls



- Once Gate has been selected from the Type dropdown menu, you will be able to choose from the list of all gates on your account via the Gate dropdown menu
- Once you have selected the gate you would like, click OK

Now we will look at the main portion of our IVR where calls that come in during normal business hours will be directed and then routed based on the caller's choice. For this example we'll assume we have three main places a caller could be routed to: Billing, Tech Support, and General Customer Support. We will also add a fourth destination that can be reached if the caller knows the extension they would like to reach. With this in mind, let's get started.

- Double-click the Get Input node
- Change Type from Min & Max to Exact input via the drop down menu (for more information about the Min & Max uses, see the Features section of this guide)
- The Get Digits node does support Speech Recognition if that is something you would like enabled, though extra charges for this feature may apply. For this case, we'll keep that feature disabled
- We can now choose our audio files we want played for the following three circumstances:
    - Audio File: will be the main audio the caller hears when they get to this part of the IVR. Something to the effect of "Press 1 for Billing, 2 for Tech support, 3 for all other questions. Or if you know your party's extension, please enter it now."
    - No Input: this will play when the caller has not entered any input in the timeframe we set in the Response Timeout section
    - Invalid Input: If the caller enters a number or set of numbers that we have not set up to do anything here (say they press 5 when we only have 1,2,3 as valid inputs)
    - Note: The audio files can be assigned in similar fashion to the way we assigned an audio file to the Play Audio node earlier



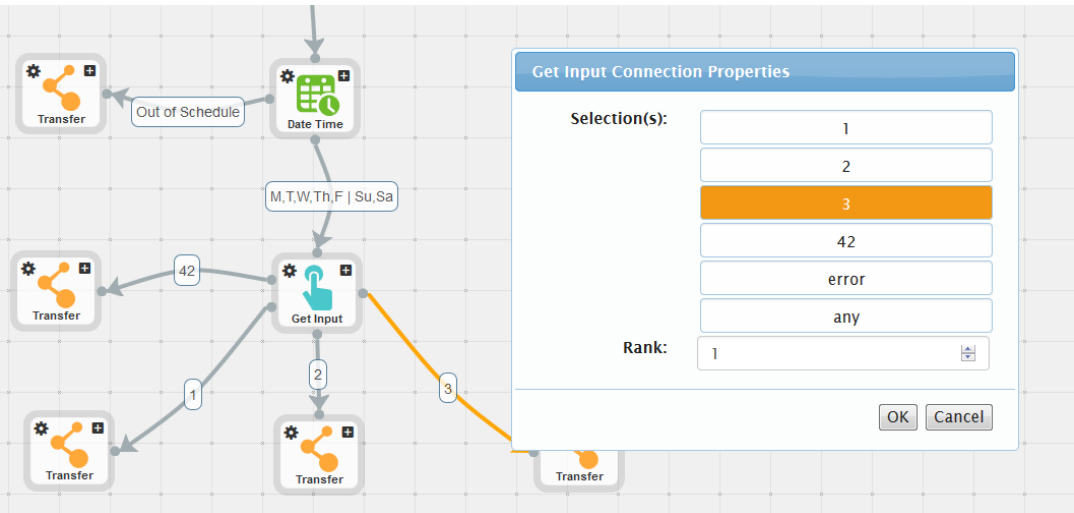
- Now let's add our acceptable inputs
    - The first input is already set for us – 1
    - To add a new input, simply click the Add Input button and type in the input you would like in the resulting field
    - For our example we will have inputs of 1,2,3 and 42 for the specific extension
- Lastly, we can set how long we want to give the caller to be able to enter their input and how many retries the IVR will give them before returning an error (which we will show how to deal with later). The default is 5s for Response Timeout and 3 Retries, though you can tailor this to your needs.
- When everything is set as you want, click OK

So we now should have our choices set for where we would like the call to go. All we need to do now is set up those transfer nodes and their connections to the Get Input node.

- Since we have four places the call can ultimately go, we will add four transfer nodes to the grid
- Add one connection from the Get Digits node to each of the four Transfer nodes
- Once the connections are made, we can edit the connection properties for each one

When you double-click on any of the connections between the Get Input and Transfer nodes you created, you will get a Get Input Connection Properties window with options specific to the Get Input properties you applied to that node.

- Notice you have several options to choose from – the options we set up earlier, and two others: error and any
- To assign the appropriate option to the corresponding Transfer node, simply click the number you want to the customer to press to go to that node. See the figure below for an idea on how you might want this to look. Keep in mind, you can have these and any of the other nodes in any layout you like as long as the connections and their properties do what is desired for your setup.



- Once the connections have been properly configured, the last step is to configure your transfer destinations as needed.

There you have it. You should now have a basic working IVR. Make sure you hit Save in the upper right-hand corner! It's a good idea to do this every once and a while as you are building your IVR anyways to ensure any changes you are making do not get lost in case of a network outage, a computer issue, etc. Obviously this does not cover even close to all of the various types of situations that may be unique to your needs for your call center, but hopefully this guide will give you a foundation of knowledge upon which you can construct your own custom IVR using the IVR Studio and the concepts learned here.

3.0 Node Reference

Node	Definitions and Properties
Start	This is where the IVR experience begins. Click and drag from the plus sign at the top right of the **Start** box to build a connection to your first node (Tool).
Answer	The purpose of the **Answer** node is to represent when the customer experience becomes engaged with the IVR system. It does not necessarily have to be the first node in the IVR. For example: A **Scripting** node could be placed prior to the Answer node. That script could then determine how the call is handled, including directing the call down one connection or another. Each connection can have its own Answer node.
Hang up	The purpose of the **Hang up** node is to initiate the end of the call within the IVR. However, connections to other nodes can be placed after the call has ended. For example: The **WWW** node could be placed after **Hang up** so that data collected from that call can be sent to a server.
Start Recording	When a call encounters this node, audio recording will be initiated.
Stop Recording	When a call encounters this node, audio recording will cease.
Transfer	This node allows calls to be transferred off of the IVR with a choice of destination type. **Node Id:** Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. **Type:** Select a transfer destination. All **Gate**, **Cloud**, and **Track** destinations must currently be configured on the same account as the IVR. ➢ **Gate:** The call will be transferred from the IVR to an inbound gate. ➢ **Cloud Profile:** The call will be transferred from the IVR to a cloud destination. ➢ **Track:** The call will be transferred from the IVR to a track destination. ➢ **PSTN:** The call will be transferred to any 10 digit domestic number with an optional extension. ➢ **SIP:** The call with be transferred to the provided SIP destination. For PSTN and SIP type transfers, the ANI, DNIS or a Custom ID can be chosen as a given Caller ID. Other options include being able to select hold music or Post Dial Audio to be played for the customer at transfer. Features are also available for the agent recieving the transfer. An Agent Whisper can be set to alert the agent and TT Accept can be enabled so the agent has to select a digit on the keypad before accepting the call. A recording option is also available for PSTN and SIP transfers, where you can explicitly start or

	stop call recording on transfer. If the Set Recording field is not checked, then the previous recording setting will continue.
Play Audio	**Play Audio** allows you to insert an audio file anywhere in the IVR. **Node Id:** Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. **Audio Files:** Audio files must be uploaded to the Audio Library. You can then choose to select the file by going to **Open Audio Library** and browsing all of the audio files available, or you can select **Manual Entry** and enter the exact name of the audio file needed (which can also be the name of a variable defined previously in the IVR Studio, prefixed by the '$' character). **Enable Barge In:** This feature allows audio to be interrupted by input.
Get Input	**Get Input** is able to request and gather numeric or verbal information from the customer. **Node Id:** Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. **Type:** ➢ **Min & Max:** This option allows you to set a range of digits. You can then enter the value for the lowest amount of digits allowed in the **Min Digits** field and the highest amount allowed in the **Max Digits** field. ➢ **Exact Input:** This option allows you to set specific numbers or words for the customer to choose from. Selection choices can be added in the **User Input** field. To give the customer multiple options, simple select **Add Input** to create more **User Input** fields. **Audio File:** This field can be used to add an audio prompt that instructs customers to give the required information. You can choose to select the file by going to **Open Audio Library** and browsing all of the audio files available, or you can select **Manual Entry** and enter the exact name of the audio file needed. **No Input:** This field can be used to add an audio prompt that plays if the customer does not respond within a set amount of time. You can choose to select the file by going to **Open Audio Library** and browsing all of the audio files available, or you can select **Manual Entry** and enter the exact name of the audio file needed (which can also be the name of a variable defined previously in the IVR Studio, prefixed by the '$' character). **Invalid Input:** This field can be used to add an audio prompt that plays if the customer responds with an invalid input. You can choose to select the file by going to **Open Audio Library** and browsing all of the audio files

	available, or you can select **Manual Entry** and enter the exact name of the audio file needed (which can also be the name of a variable defined previously in the IVR Studio, prefixed by the '$' character). **Speech Rec. Type:** This feature allows the IVR to recognize verbal input from the caller. ➢ **Numeric Only** ➢ **Alpha Only** ➢ **Alpha-Numeric** **Min Length:** Set the maximum time, in seconds, that the caller has to supply his or her input. **Max Length:** Set the minimum time, in seconds, that the caller has to supply his or her input. **Response Timeout:** Use this field to set the time, in seconds, within which the customer has to respond. **Retries:** Use this field to enter the number of retries the customer has to make a valid entry. **Get Input Connection Properties:** When a connection is created from the **Get Input** node to any other node, a bubble appears between the two. Double click on this bubble to specify which customer input routes the call down this specific connection. **Error:** This can be set up to help direct calls in which no entry or an entry outside of the set parameters has been made. **Any:** This will direct calls in which any input has been given. **Rank:** A rank can be assigned to connections to determine priority in how the call is handled when the input given could match the criteria of more than one Connection Property.
Say Variable	This node allows the system to verbally repeat a specified value. **Node Id:** Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. **Type:** The following options determine the format in which the identified value will be repeated. ➢ **sayDigits:** Each digit will be said as an individual number. For example, the value "1357" will be spoken as "one-three-five-seven". ➢ **sayDate:** The value must be in the format YYYY-MM-DD to be repeated accurately as a date. ➢ **sayNumber:** The value will be repeated as one number. For example, the value "1357" will spoken as "one thousand three hundred and fifty-seven" ➢ **sayAlpha:** Each letter will be said as an individual character. For example, "fox" will be spoken as "f-o-x" ➢ **Text-to-Speech:** A text value will be spoken as a complete word or phrase (?). For example, "fox" will be spoken as "fox".

	Intro Audio: This audio will play directly prior to the given value. You can choose to select the file by going to **Open Audio Library** and browsing all of the audio files available, or you can select **Manual Entry** and enter the exact name of the audio file needed (which can also be the name of a variable defined previously in the IVR Studio, prefixed by the '$' character). **Say Value:** In this field enter the Node id in which the value you wish to be repeated has been provided. The Node id must be proceeded by a '$' character. For example, "$getDigits_1". You may also specify a variable previously defined in the IVR Studio by prefixing the variable name by '$' – for example, "$myVariable. **Closing Audio:** This audio will play directly following the given value. You can choose to select the file by going to **Open Audio Library** and browsing all of the audio files available, or you can select **Manual Entry** and enter the exact name of the audio file needed (which can also be the name of a variable defined previously in the IVR Studio, prefixed by the '$' character).
Record Audio	This node allows customer responses to be recorded and saved as an individual audio file. **Node Id:** Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. **Record Prompt:** Enter an audio file prompting the caller to state the desired information. You can choose to select the file by going to **Open Audio Library** and browsing all of the audio files available, or you can select **Manual Entry** and enter the exact name of the audio file needed (which can also be the name of a variable defined previously in the IVR Studio, prefixed by the '$' character). **File Name:** Audio recordings will be saved as a .wav file. The UII will automatically be pulled and used as the beginning of the file name. A field is also provided for a unique file name to be entered after the caller id. **Enable Append Audio:** If Append Audio is enabled, then the audio from multiple nodes can be combined into just one audio file. The audio recordings you wish to append will need to share the same filename. If Append Audio is not enabled for all audio recordings sharing the same name, then subsequent recordings will simply replace previous ones. **Enable Play Beep:** With this feature enabled, a beep will prompt the customer when the audio recording starts. **Max Duration:** Set the maximum time, in seconds, that the caller has to complete their recorded statement.
Date Time	This node allows schedules to be applied to connections. Adding schedules will determine how calls are handled at specific times.

	Node Id: Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. **Timezone:** Select a timezone for the schedule. **<u>Date Time Connection Properties:</u>** Click and drag from the plus sign on the top right of the Date Time node to the next node in the IVR. A properties bubble will appear on the connection between the two nodes. Double click the bubble to customize the schedule. Select either **Out of Schedule** or **New Schedule**. When **New Schedule** is selected, you will be able to check the box next to the day or days you want to be in-schedule for that connection. **Time Range** then allows you to create multiple in-schedule time blocks by selecting the plus sign on the grey slider scale. By sliding the white boxes, or typing a time (hh:mm AM/PM) in the text boxes below the slider, you can then increase or decrease the time in the range. Click **Apply** to save that schedule. Multiple schedules can be attributed to a connection by again clicking **New Schedule** within the **Date Time Connection Properties** window. If set to **Out of Schedule**, this connection will be the default path for any call arriving outside of a scheduled day or time. Multiple connections can also be built from one Date Time node to other nodes.
Scripting	**Node Id:** Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. You are able to add JavaScript in the text area of this node to implement complex logic and decision making or branching. **Connection Id:** Using javascript in the Scripting node you can designate which connection path should be followed when exiting the Scripting node. The syntax for designating which exit connection to take is: `ivr.setConnection("connection_id");`
WWW	This node allows the IVR to interact with web services. **Node Id:** Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. **Type:** Chose the protocol for interacting with your web service. ➢ **HTTP Get** ➢ **HTTP Post** ➢ **HTTP Put** ➢ **SOAP** **URL:** The web address for the service. **Timeout:** The amount of time allowed before a timeout error is returned from an unresponsive service.

	Variables: When using HTTP Get you can specify any variables needed to call the webservice. Supply the Name of the variable as expected by the webservice, the variable Type, and where the data will be retrieved from within the IVR Studio with the Mapping field. **Format:** When using HTTP Post/Put you can specify the format in which the data will be sent. ➢ URL Values – variables are sent to the webservice in the url string ➢ XML – data is sent as xml in the body of the HTTP Post/Put ➢ JSON – data is sent as JSON in the body of the HTTP Post/Put ➢ Plain Text – data is sent as plain text in the body of the HTTP Post/Put **SOAP:** For SOAP calls, once you enter the URL for the webservice, click "Parse WSDL" to auto populate a table with the required variables for the SOAP service. Then select the Type for each variable and where the data will be retrieved from within the IVR Studio with the Mapping field. If using an IVR Variable, you do not need to prefix the variable name with '$'.
Email	This node triggers an email to be sent. Variables predefined in the IVR Studio may be used in any of the fields, excluding the Node Id field (e.g. To, CC, BCC, From, Subject, and Body Content). To include a variable, prefix the variable name with the '$' character. For example, if you have previously defined a variable called '`emailTo`' in a script node using: `ivr.putData("email1@gmail.com, email2@gmail.com","emailTo")` Then you can include this variable in the Email Node's To field as `$emailTo` **Node Id:** Each node can be assigned a unique Node Id. This Id allows that node to be identified and to interact with other nodes and features within the IVR studio. **To:** The email address, or comma separated list of email addresses, where the message will be sent. **CC/BCC:** Standard carbon copy and blind carbon copy address fields. A single email address or comma separated list of email addresses expected. **From:** A single email address that will be displayed as the sender of the message. **Subject:** The text that will be displayed in the Subject line of the email. **Body Content:** The email message content.

4.0 Scripting

The script node code block provides a JavaScript sandbox that allows any valid standalone JavaScript code to be run as an IVR is processing. For security reasons, native JavaScript that allows for network requests have been disabled.

4.1 ivr.stringToXml

Attempts to parse the provided string as an XML document and returns the Document. See (Other Examples) for information on manipulating a Document object.

4.1.1 Parameters

- XML – A string of XML to parse.

4.1.2 Returns – An org.w3c.dom.Document Java object. Please see the relevant Java documentation for detailed information on how to interact with this object.

4.2 ivr.xmlToString

Returns the String representation of a org.w3c.dom.Document object.

4.2.1 Parameters

- XML – The Document object to process.

4.2.2 Returns – A String representation of the Document.

4.3 ivr.getNodeByUid

Returns a Node from the IVR with the provided UID in the form of a JSON object. These nodes are a common parameter for other functions in the IVR scope.

4.3.1 Parameters

- UID - The UID String of the node to fetch.

4.3.2 Returns – A JSON object representation of a node that can be used as a parameter for other API calls.

4.4 ivr.executeHttpNode

This can be used to execute WWW node without needing to leave the scope of the script block.

4.4.1 Parameters

- Node - The HTTP type Node object to execute.

4.4.2 Returns – A Node object that the IVR would execute next if the HTTP node had been executed normally.

4.5 ivr.setDateFormat

Sets the Date format used by the formatDate function. The default format is "yyyy-MM-dd HH:mm:ss". For a list of valid characters and their meanings, see the documentation on java.text.SimpleDateFormat.

4.5.1 Parameters

- Format – The format String to apply.

4.6 ivr.formatDate

Returns the String representation of the given Date object. The format can be set with the setDateFormat function.

4.6.1 Parameters

- Date – The date to format.

4.6.2 Returns – A formatted string representation of the provided date.

4.7 ivr.getUii

Used to get the UII, a unique identifier of the current call in the CCI platform.

4.7.1 Returns – A string representing the call UII.

4.8 ivr.getAccountId

Returns the Account ID that the current call belongs to.

4.8.1 Returns – A string representing the call's Account ID.

4.9 ivr.getAni

Returns the caller's phone number.

4.9.1 Returns – An unformatted String of the ANI.

4.10 ivr.getDnis

Returns the number the CCI platform received the call on

4.10.1 Returns – An unformatted String of the DNIS.

4.11 ivr.getRecordingUrl

Returns the location where the call recording will be available after the call is terminated and processed.

4.11.1 Returns – A String containing the eventual URL of the call recording.

4.12 ivr.getData

Fetches data that was added to the key/value store by the putData() function.

4.12.1 Parameters

- Key – Name of the data point to retrieve.

4.12.2 Returns – The value associated with the key.

4.13 ivr.putData

Adds data to a key/value store that is unique to the current call. This data is not persistent and is discarded at the end of the call. Use the putReportField() function if you wish to add data that will persist after the call terminates and be available in reports.

4.13.1 Parameters

- Key – A name for the data point.
- Value – The value to associate with the Key.

4.14 ivr.getReportField

Fetches data that was added to the reportable key/value store by the putReportField() function.

4.14.1 Parameters

- Key – Name of the data point to retrieve.

4.14.2 Returns – The value associated with the key.

4.15 ivr.putReportField

Adds data to a key/value store that is unique to the current call and persists after the call completes. This data can be retrieved with the IVR Studio Call Detail report.

4.15.1 Parameters

- Key – A name for the data point.
- Value – The value to associate with the Key.

4.16 ivr.setConnection

Sets the name of the connection to the next node that will be processed when the current script ends.

4.16.1 Parameters

- Connection – The ID of a connection from this script node.

4.17 ivr.getConnection

Gets the name of the connection to the next node that will be processed when the current script ends.

4.17.1 Returns – The ID of the connection currently set to be processed next.

4.18 ivr.sleep

Pauses execution for the specified number of milliseconds.

4.18.1 Parameters

- milliseconds – Duration to sleep.

4.19 ivr.debug

Adds a message to the debug log that can be delivered to a specified e-mail address when the IVR call completes.

4.19.1 Parameters

- message – String to include in the log.