



The Intelligent Cloud Contact Center

Interactive Voice Response (IVR)

Administrator's Guide

April 2026

Learn how to configure Interactive Voice Response (IVR) scripts, which are visual devices for routing calls in inbound and autodial campaigns.

Five9 and the Five9 logo are registered trademarks of Five9 and its subsidiaries in the United States and other countries. Other marks and brands may be claimed as the property of others. The product plans, specifications, and descriptions herein are provided for information only and subject to change without notice, and are provided without warranty of any kind, express or implied. Copyright © 2026 Five9, Inc.

About Five9

Five9 is the leading provider of cloud contact center software, bringing the power of the cloud to thousands of customers and facilitating more than three billion customer interactions annually. Since 2001, Five9 has led the cloud revolution in contact centers, delivering software to help organizations of every size transition from premise-based software to the cloud. With its extensive expertise, technology, and ecosystem of partners, Five9 delivers secure, reliable, scalable cloud contact center software to help businesses create exceptional customer experiences, increase agent productivity and deliver tangible results. For more information visit www.five9.com.

Trademarks

Five9®

Five9 Logo

Five9® SoCoCare™

Five9® Connect™

Contents

What's New	14
Interactive Voice Response	15
Visual IVR Scripts	16
Benefits	16
Implementation	17
Reports	17
Use Cases	17
Contacting Customer Support	18
Creating Multilingual Elements	19
Planning Your Implementation	20
Evaluating Your Current Strategy	20
Assigning Implementation Roles	21
Testing the Visual IVR	21
Releasing the Visual IVR	22
Preparing Your Inbound Campaigns	22
Enabling the Visual Mode	23
Enabling National and International Callbacks	23
Securing Your Visual IVR Pages	23
Using the Visual IVR URL	25
Terminating the Visual IVR	25
Integrating a Visual IVR Script in Your Web Site	26
Customizing the Script	26
Adding a Button or Link to a Web Page	27
Customizing the Visual IVR Page	27
Preparing Visual IVR Scripts	29
Choosing Modules	29
Available Modules	30
Modules with Collapsible Prompts	30
Modules that Accept Images	30
Creating Visual IVR Scripts	32
Input Module	32
Lookup Contact Record Module	35
First Menu Module	35
Skill Transfer Modules	38
Second Menu Module	38
Phone Branch	39
Chat Branch	40
Email Branch	41
Converting Voice- and Digit-Based Scripts into Visual IVR Scripts	41
Five9 Engagement Workflow	43
Configuring Engagement Workflow Templates	43

Implementing IVR Scripts	45
Managing IVR Scripts	45
Creating IVR Scripts	46
Duplicating IVR Scripts	46
Creating IVR Templates	47
Configuring IVR Modules	49
Adding and Removing Modules	50
Selecting Icons	51
Connecting Icons	51
Removing Connections	52
Clearing the Work Area	52
Updating IVR Scripts	53
Exporting and Importing IVR Scripts	53
Exporting IVR Scripts	54
Importing IVR Scripts	55
Backing Up IVR Scripts	56
Restoring IVR Scripts	57
Scheduling IVR Scripts in Campaign Properties	57
Managing Variables	59
Script Variables	59
Adding Variables	60
Editing and Deleting Variables	62
Call Variables	63
Adding Functions	64
Adding Prompts	66
Adding Prompts	67
Selecting Dispositions	68
Using Recorded Files	69
Importing Scripts	70
 Designing IVR Scripts	 72
IVR Script Designer	72
Actions Menu	75
Script Menu	76
Default Properties	77
Variables	79
Languages	82
Multilingual Elements	82
Speech Recognition Grammars	86
alphanum	86
boolean	90
ccexpdate	91
creditcard	92
currency	93

date	94
digits	95
number	96
phone	97
socialsecurity	97
time	98
zipcode	98
IVR Modules	99
Incoming Call Module	100
Hangup Module	100
Start on Hangup Module	102
Troubleshooting an IVR Script	103
Modules	104
Agent/Voice Mail Transfer Module	104
Agent Assignment for Text Channel Interactions	104
Transferring to a Specific Agent	105
Transferring to Any Available Agent	105
Agent/Voicemail Assignment for Voice Interactions	105
General Tab	106
Agent Tab	107
Voice Mail Tab	108
Prompts Tab	109
Dispositions Tab	109
Answering Machine Module	109
Case Module	111
Conference Module	113
General Tab	114
Participants Tab	114
Prompt Tab	116
Dispositions Tab	117
Contact Update Module	118
SFDC Einstein Module	119
Properties	122
General Tab	123
Parameters Tab	123
Example IVR Scripts with SFDC Einstein Module	125
Extension Transfer Module	125
General Tab	126
Dispositions Tab	126
Five9 Rules Engine Module	127
General Tab	130
Parameters Tab	130
Foreign Script Module	132

General Tab	133
Parameters Tab	133
Get Digits Module	135
General Tab	136
Prompts Tab	138
Page Tab	138
Text Tab	142
Dispositions Tab	143
If/Else Module	143
General Tab	144
Input Module	146
Digital Channels	146
Languages	147
Properties	147
General Tab	148
Prompts Tab	152
Events Tab	152
Confirmation Tab	155
Page Tab	156
Text Tab	160
Iterator Module	161
IVA Transfer Module	162
Properties	162
General tab	163
Authorization Tab	164
Disposition Tab	165
Chat Customer Disconnect or Timeout	165
Using the Chatbot Script	165
Creating the IVR Script Variables	166
Looking up Contact Information	167
Storing and Using Your Contact Information	168
Sending the Interaction to the Chatbot	168
Storing the Returned Variable Information	169
Continuing the Interaction	170
Adding New Contact Information	170
Sending the Interaction to an Agent	171
Ending the Interaction	171
Language Module	173
General Tab	173
Prompts Tab	174
Events Tab	176
Dispositions Tab	178
Lookup Contact Record Module	178
General Tab	179

Prompts Tab	182
Page Tab	182
Menu Module	185
Digital Channels	185
Properties	186
General Tab	187
Branches Tab	190
Prompts Tab	191
Events Tab	192
Confirmation Tab	194
Page Tab	195
Dispositions Tab	198
Natural Language Processing Module	198
Properties	201
General Tab	202
Parameters Tab	202
Example IVR Scripts with Five9 NLP Module	203
Play Module	204
Properties	204
General Tab	204
Prompts Tab	205
Page Tab	206
Dispositions Tab	209
Email Reply Tab	209
Text-to-Speech Say As Options	211
Query Module	214
Types of Database	214
Properties	215
General Tab	215
Request Tab	216
Response Tab	217
Prompts Tab	220
Page Tab	221
Text Tab	223
Use Case	224
Preparing Your Database	225
Creating Script Variables	226
Configuring the Query Module	227
Customizing the Script	230
Testing Your Implementation	233
Five9 Salesforce Module	233
Properties	234
General Tab	234
Parameters Tab	235

Example IVR Scripts with Salesforce Module	236
Set DNC Module	237
Set Variable Module	238
Converting Integers	244
Using the KVLIST Data Type Functions	246
Defining a KVLIST Collection	247
Using KVLIST Variable Functions	247
Skill Transfer Module	250
General Tab	252
Skills Tab	254
Announcements Tab	255
Estimated Wait Time	258
Priority Tab	259
Connectors Tab	260
Queue Callback Tab	260
Dispositions Tab	262
Disposition Flow	263
System Info Module	264
Properties	265
General Tab	265
Prompts Tab	265
Page Tab	266
Defining Search Conditions	268
Defining Returned Parameters	272
Third Party Transfer Module	274
General Tab	274
Dispositions Tab	276
Update System Info Module	276
Voice Input Module	277
General Tab	278
Prompts Tab	280
Events Tab	281
Confirmation Tab	282
Page Tab	283
Dispositions Tab	287
Voicemail Transfer Module	287
General Tab	288
Dispositions Tab	289
IVR Examples	290
Simple Campaign Script for Post-Call Surveys	290
Get Digits Module	291
If/Else Module	291
Set Variable Module (IF Branch)	292

Set Variable Module (ELSE Branch)	293
Complex Campaign Script for Post-Call Surveys	294
Call Variables	294
IVR Script	297
Creating an IVR Script	300
Adding and Configuring the Play Module	303
Adding and Configuring the Get Digits Module	304
Routing Users to Skill Groups	305
Configuring the Sales Branch	308
Configuring the Support Branch	314
Configuring the No Match Branch	314
Adding the Hang Up Module	315
Adding and Configuring the If/Else Module	316
Adding and Configuring the Set Variable Module	318
Linking From the If/Else Set Variables Modules	320
Creating an IVR Loop	321
Defining a Variable	322
Adding and Configuring the Get Digits Module	323
Adding and Configuring the Case Module	324
Adding and Configuring the If/Else Module	325
Adding and Configuring the Set Variable Module	326
Linking the Modules	329
Configuring the Query Module	329
Creating an IVR Script with XML Web Service Query	332
Input Module	332
Query Module	333
Play Module	334
Iterator Module	335
Creating an IVR Script with Queue Callback	336
Call Variables Configuration	336
IVR Script Configuration	337
Set Variable Module	337
Skill Transfer Module	338
Creating an IVR Script for Tracking a Shipment	340
Creating Variables to Support Query Responses	340
Adding and Configuring the Query Modules	341
Configuring the GetInitialShipmentInfo Query Module	341
Adding and Configuring the IfDelivered If/Else Module	344
Adding and Configuring the ActualDeliveryInfo Branch	344
Adding and Configuring the GetEstimatedDeliveryInfo Branch	346
Adding and Configuring the ReturnVariableDebug Module	346
Adding and Configuring the IfErrorDebug Module	348
Using an Agent Extension in an IVR	348
Configuring the Get Digits Module	349

Configuring the Case Module	349
Configuring the Agent/Voice Mail Transfer Module	351
Adding Modules for Each Case	352
Connecting the Modules	353
Incoming Call Transfer to Studio IVA	357
Preparing Your Script Variables	357
Collecting Data for IVA (to Studio)	358
Sending the Interaction to Studio IVA	360
Data Return (From Studio)	361
Managing Voicemail	362
Retrieving Voicemail Messages	362
Get Digits Module	362
System Info Module	363
Get Digits Module	364
If/Else Module	365
If/Else Module	366
System Info Module	366
Play Module	367
Update System Info Module	367
Get Digits Module	368
If/Else Module	369
Update System Info Module	370
New Iterator Module	370
Configuring Voicemail Account for New Users	371
If/Else Module	371
Get Digits Module	372
Update System Info Module	373
Play Module	374
Get Digits Module	375
Play Module	375
Playing the Current Voicemail Greeting	376
If/Else Module	377
Play Module	377
Play Greeting Module	378
Recording a Voicemail Greeting	378
Case Module	379
Voice Input Module	379
Play Module	381
Changing an Existing Voicemail PIN	381
Get Digits Module	381
Update System Info Module	382
Play Module	383
Creating an Opt-Out IVR Script	383
Creating a Variable to Track Calls and Numbers Added to the DNC List	384

Creating the IVR Script	385
Get Digits Module	385
If/Else Module	387
Set Variable Modules	387
Set DNC Module	388
Play Module	388
Hangup Modules	389
Configuring the Outbound Campaign Properties	390
Testing Your IVR Script and Outbound Campaign	392
Flowcharts	393
Error Messages	395
AI Agent Connect	400
Cresta.AI Implementation Guide	400
Prerequisites	400
Integration Process	401
What Cresta.AI Does (Informational Only)	401
What You Do (Customer Responsibility)	401
What the Five9 Implementation Team Does (Informational Only)	401
IVR Transfer Configuration	402
Header Usage	407
Headers Sent to the IVA	407
Headers Received from the IVA	408
Decagon Implementation Guide	408
Prerequisites	408
Integration Process	409
What Decagon Does (Informational Only)	409
What You Do (Customer Responsibility)	409
What the Five9 Implementation Team Does (Informational Only)	409
IVR Transfer Configuration	410
Header Usage	414
Headers Sent to the IVA	414
Headers Received from the IVA	415
FlipCX Implementation Guide	415
Prerequisites	415
Integration Process	416
What FlipCX Does (Informational Only)	416
What You Do (Customer Responsibility)	416
What the Five9 Implementation Team Does (Informational Only)	416
IVR Transfer Configuration	417
Header Usage	422
Headers Sent to the IVA	422
Headers Received from the IVA	422

Hyro Implementation Guide	423
Prerequisites	423
Integration Process	423
What Hyro Does (Informational Only)	423
What You Do (Customer Responsibility)	424
What the Five9 Implementation Team Does (Informational Only)	424
IVR Transfer Configuration	425
Header Usage	431
Headers Sent to the IVA	431
Headers Received from the IVA	431
Kastle.AI Implementation Guide	432
Prerequisites	432
Integration Process	432
What Kastle.AI Does (Informational Only)	432
What You Do (Customer Responsibility)	433
What the Five9 Implementation Team Does (Informational Only)	433
IVR Transfer Configuration	433
Header Usage	439
Headers Sent to the IVA	439
Headers Received from the IVA	440
LevelAI Implementation Guide	440
Prerequisites	440
Integration Process	441
What LevelAI Does (Informational Only)	441
What You Do (Customer Responsibility)	441
What the Five9 Implementation Team Does (Informational Only)	441
IVR Transfer Configuration	442
Header Usage	445
Headers Sent to the IVA	445
Headers Received from the IVA	446
Parloa.AI Implementation Guide	446
Prerequisites	446
Integration Process	447
What Parloa.AI Does (Informational Only)	447
What You Do (Customer Responsibility)	447
What the Five9 Implementation Team Does (Informational Only)	447
IVR Transfer Configuration	448
Header Usage	453
Headers Sent to the IVA	453
Headers Received from the IVA	454
Sierra Implementation Guide	454
Prerequisites	454
Integration Process	455
What Sierra Does (Informational Only)	455

What You Do (Customer Responsibility)	455
What the Five9 Implementation Team Does (Informational Only)	455
IVR Transfer Configuration	457
Header Usage	461
Headers Sent to the IVA	461
Headers Received from the IVA	462

What's New

Release	Change	Topic
Apr 2026	Added new implementation guides.	Decagon Implementation Guide Hyro Implementation Guide Sierra Implementation Guide LevelAI Implementation Guide
Mar 2026	Updated several implementation guides to improve accuracy and clarity.	Cresta.AI Implementation Guide FlipCX Implementation Guide Kastle.AI Implementation Guide
	Added new Parloa.AI implementation guide.	Parloa.AI Implementation Guide
Feb 2026	Added new topic on AI Agent Connect that integrates your Five9 IVR system with certified third-party AI providers.	AI Agent Connect
Nov 2025	Added error response note.	Third Party Transfer Module
Sep 2025	Updated the Disposition tab image in the Hangup Module.	IVR Modules Creating the IVR Script
Jun 2025	Updated navigation and text in introduction paragraph.	Scheduling IVR Scripts in Campaign Properties

Interactive Voice Response

Interactive Voice Response (IVR) enables human-computer interaction for inbound (required) and autodial (optional) campaign calls. IVR scripts can range from simple announcements to complex multi-prompt customer service menus that can perform many functions, such as these:

- Route incoming calls according to the availability of agents or skill groups
- Transfer calls to a third party or voicemail when agents are not available
- Provide instructions to callers
- Search for contact records
- Assign data to variables

Interactive Voice Response is available in these options:

- Visual mode: Visual IVR is an optional feature in your Virtual Contact Center. For more information, contact your Five9 representative.

The Visual IVR feature is a self-service interface that enables you to provide on your Web site a visual interface on any device, such as a smartphone. This interface enables your callers to click or touch menu items without having to listen and respond verbally to each option. To interact with an agent, callers can choose among the contact options that you offer, such as phone, Five9 Email, and Five9 Chat. To integrate Visual IVR in your workflow, see [Visual IVR Scripts](#).

- Voice mode: The conventional voice mode enables you to use recorded messages as prompts and to create prompts specific to your needs. You can use a text-to-speech builder and speech recognition. For more information, see [Adding Prompts](#) and [Speech Recognition Grammars](#).
- Touch-tone keypad mode: The default mode enables customers to use a touch-tone phone to interact with your company.

You can use a script in several campaigns, you can link small scripts to create a large one, and you can use several scripts in the same campaign. You can export and import IVR scripts to create a customized version for a different campaign. You can access IVR information in reports. IVR scripts can contain diverse components, such as speech recognition, linked modules, and variables.

Visual IVR Scripts

Five9 Visual IVR is an optional feature of your Virtual Contact Center that provides to your mobile customers an alternative to voice-only IVR flows. For more information, contact your Five9 representative.

[Benefits](#)

[Implementation](#)

[Reports](#)

[Use Cases](#)

[Planning Your Implementation](#)

[Preparing Your Inbound Campaigns](#)

[Integrating a Visual IVR Script in Your Web Site](#)

[Preparing Visual IVR Scripts](#)

With Visual IVR, you can convert complex voice IVR prompts into a convenient visual flow that improves customer experience. You can use Visual IVR scripts in many cases, such as these:

- Viewing and paying a bill.
- Obtaining information, such as searching an FAQ.
- Requesting customer support.
- Changing hotel or restaurant reservations.
- Associating skill groups with campaigns to transfer Visual IVR transactions to the chat sessions available to the skill groups.
- Pass data, such as location, customer's information, and product information

Benefits

When callers use a mobile device, the convenience of navigating through such a visual IVR flow results in these many benefits:

- For your business:
 - Self service results in fewer calls when you implement Visual IVR on Web pages that customers reach to obtain support, such as billing and FAQ pages.
 - Fewer calls are abandoned when compared to traditional voice IVR flows.

- Fewer calls are transferred to agents because callers can provide precise context due to the choices available in the IVR flow.
- First-call resolution is increased because agents can communicate rapidly and efficiently with customers by callback, Five9 Chat, or Five9 Email.
- Style and language in the Visual IVR are consistent with your brand and customer satisfaction criteria.
- For callers:
 - Self service is available at all times regardless of the device and location.
 - Callers can choose the contact method by your agents without waiting on hold when the voice option is not convenient.

Implementation

Visual IVR is simple to implement and manage by using the same drag-and-drop tools that you are already using for voice IVRs. Depending on your configuration, Visual IVR may include estimated wait time, callbacks, Five9 Chat, Five9 Email, and Five9 Visual Customer Feedback. At the end of the Visual IVR flow, Five9 Visual Customer Feedback helps you to measure customer satisfaction when the customer's sentiment is most relevant and accurate.

Reports

Easy-to-use survey options and out-of-the-box reports enables you to monitor how your customers feel and resolve problems. With real-time and historical reports, administrators and supervisors can easily monitor the results and refine the Visual IVR flows.

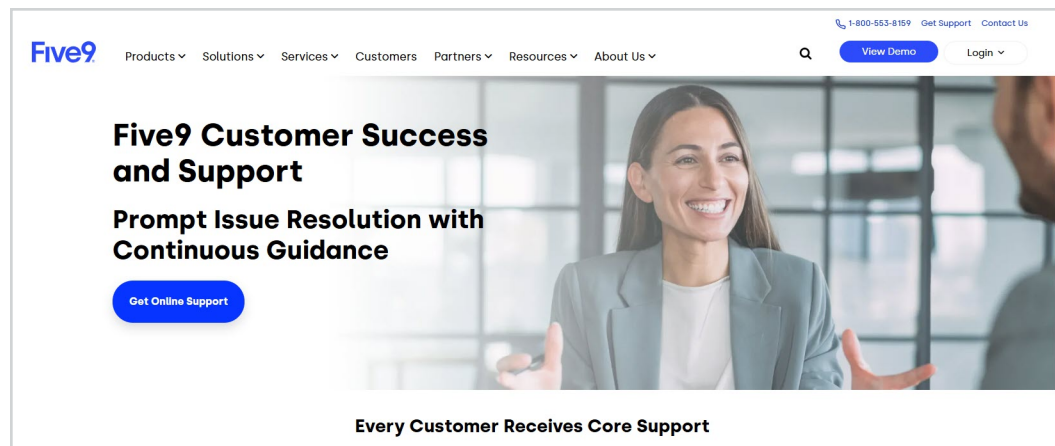
Use Cases

- [Planning Your Implementation](#)
- [Preparing Your Inbound Campaigns](#)

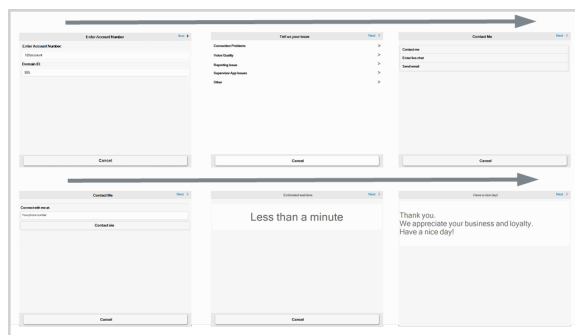
- [Preparing Visual IVR Scripts](#)
- [Preparing Visual IVR Scripts](#)

Contacting Customer Support

This example shows a typical Visual IVR flow in which a customer contacts Five9 to discuss an issue with an agent. In this case, the caller choose to be contacted by phone by an agent.



After clicking the contact button, the caller is led through a series of visual prompts. Each figure below shows how the caller progresses rapidly through the Visual IVR. When the caller closes the window, an agent who has the appropriate skill will return the call within one minute. Each IVR page is displayed on the entire screen of the customer's device.



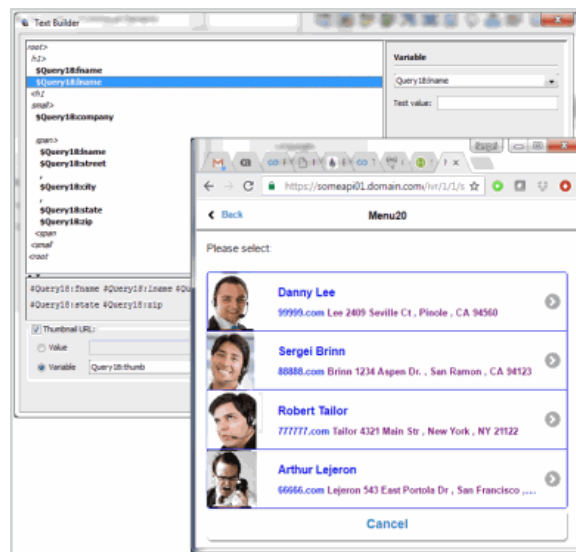
For the IVR modules used to create this flow, see [Creating Visual IVR Scripts](#).

Creating Multilingual Elements

You can combine elements to offer dynamic menu options, such as language preference or a custom list of routing options based on skill preference or customer status. A custom tag enable you to render your desired look and feel based on the JSON response received from these Visual IVR modules:

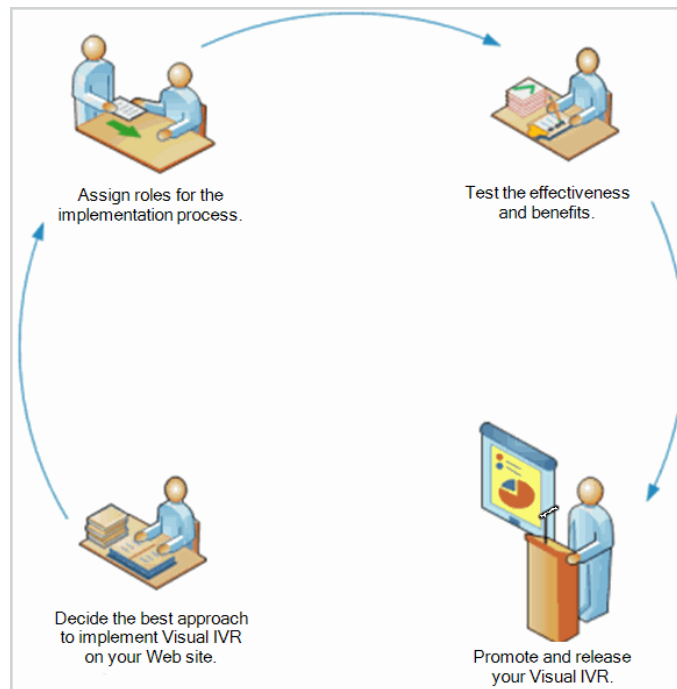
- Get Digits
- Play
- Input
- System Info
- Lookup
Contact
Record
- Voice Input
- Menu

The graphic below lists the process flow for creating and using these Visual IVR features to create multilingual elements and the dynamic menu options listed above for your Visual IVR routing solutions.



Planning Your Implementation

Before creating IVR scripts, you need to evaluate your strategy so that you can maximize the benefits of Visual IVR. This figure shows the steps, starting from the lower left.



Evaluating Your Current Strategy

To decide the best approach to implement Visual IVR on your Web site, you need to review your current self-service and mobile strategy. Start by answering these questions, but be sure to add any others that are important to your brand and customer satisfaction criteria:

- What are your goals?
- Do you use a self-service voice IVR?
- Do you have usage data about customer visits to your Web site?

- Do multiple groups in your company use or are considering using a Web self-service IVR?

Once you know the answers, you can create an overall plan.

Assigning Implementation Roles

Visual IVR is hosted by Five9. You only need to enable campaigns and configure your Web site. Several members of your organization need to collaborate in the implementation:

- VCC administrator:
 - Creates scripts specifically for Visual IVR or modifying existing voice IVR scripts. For more information, see [Preparing Visual IVR Scripts](#).
 - Enable new or existing campaigns for Visual IVR as follows:
 - Associate the IVR script in the campaign.
 - Set the number of ports available for the Visual IVR.
 - Configures one or more schedules. For more information, see [Preparing Your Inbound Campaigns](#).
- Branding and Marketing:

These teams decide the look and feel of the Visual IVR and its location in your Web site. You may choose among several themes and modify these themes if needed.
- Product Development:

This team integrates the Visual IVR in your Web site. For more information, see [Integrating a Visual IVR Script in Your Web Site](#).

Testing the Visual IVR

Before releasing the Visual IVR, you need to test its effectiveness. On the Internet, you can find tools that can help you to understand the value of a Visual IVR and how to optimize it for your business.

- 1 Add access points, such as a links or buttons, in multiple locations on your Web site.

2 Compare the results of the access points.

Start by using these criteria, but be sure to add any others that are important to you, for example:

- Most clicks, which indicates the highest number of started self-service flows.
- Most completed self-service flows.
- Most agent callback, chat, or email requests.
- Most positive feedback.

3 Use the results to set one or more access points for the production release.

Releasing the Visual IVR

When you are ready to release the Visual IVR into production, promote, release, and monitor the results.

1 Promote the new self-service feature.**2** Capture feedback and analyze reports.**3** Use the feedback that you obtain:

- Target specific customers.
- Refine your business processes, Web site design, and Visual IVR scripts.
- Train your employees to interact effectively with callers who are using a visual flow.

Preparing Your Inbound Campaigns

This section describes the required campaign configuration. To set these options, see *Creating IVR Schedule Rules* in the [Campaign Administrator's Guide](#).

Enabling the Visual Mode

To enable an inbound campaign, check Visual Mode in the IVR schedule of the campaign properties. Depending on your campaign configuration, your customers may be able to choose between voice or Visual IVR paths. When a campaign is configured to use voice and visual prompts, visual prompts take precedence over voice prompts as follows:

- If you enable Visual IVR, the HTML text entered in the Visual IVR Page tab of a module that supports visual prompts overrides all other prompts.
- If Visual IVR scripts do not exist, standard prompts are rendered visually in HTML unless the campaign contains only recorded prompts files, which cannot be rendered visually.

Enabling National and International Callbacks

To enable your agents to return calls associated with a Visual IVR, you must enable at least the first feature.

- 1 To enable the callback option in Visual IVR sessions, check **Visual Mode** in the IVR schedule of the campaign properties.

By default, the Callback option is enabled. You may now return only in-country calls.

- 2 To return international calls associated with IVR and Visual IVR sessions, check **Allow collection and callback of international numbers** in the Queue Callback Tab of the Skill Transfer module (see [Queue Callback Tab](#)).

Depending on your business, callers may request a callback to an international number. To reduce phone charges, this option enables you to select whether to include or exclude international phone numbers from callback options. If you do not enable this option, you cannot return international calls.

Securing Your Visual IVR Pages

A Cross-Site Request Forgery (CSRF) attack occurs when a link or script is added to a page in a Web site where the user has been authenticated. Because the user is trusted, the attack is insidious. The element or page contains a link or script that, if clicked, redirects your customers away from your site. For more information about this subject,

see [Clickjacking Defense Cheat Sheet](#). For information about the supported browsers, see *Application Notes* in the [Technical Requirements Reference Guide](#).

The Five9 Web Page Security feature prevents malicious pages from being embedded in a frame in your Web site. By restricting access to your Web pages, you can prevent unauthorized commands, which protects your customers and your business.

Important

To prevent malicious framing of a Web page, you must use the `X-Frame-Options` HTTP header with the value appropriate for your business. The command is ignored if you place it in an `http-equiv meta` element of the page.

Example

Sarah is starting a Visual IVR session at your Web site. She enters required information in text fields. In another browser tab, Sarah opens a page at another Web site. She does not know that this page is harmful. On that page, she clicks a fake element, which sends a request to cancel the Visual IVR session. The session is canceled. Sarah cannot receive the service that she expected from your company.

In the IVR schedule properties, select an X-Frame-Options header option to allow the correct page to be displayed in a frame:

- **DENY:** Cannot be displayed even if the frame is located in the same domain as the page. You cannot start a Visual IVR script with this value in the iframe. Use this value if you do not need to place a Visual IVR link in an iframe. This option provides the most security from attacks.
- **SAME_ORIGIN:** Can be displayed only if the frame is located in the same domain as the page. This option is mostly for internal uses because customers cannot access internal Web pages.

Example

The HTTP header of `http://shop.example.com/confirm.asp` contains `X-FRAME-OPTIONS: SAME_ORIGIN`. Any frame in the `http://shop.example.com` domain can be displayed.

- **ALLOW_FROM:** Can be displayed only if the frame is located in the domain that you specify in the field.

Example

The HTTP header of `http://shop.example.com/confirm.asp` contains
`X-FRAME-OPTIONS: ALLOW_FROM https://partner.affiliate.com.`
The page may be framed only by pages in the
`https://partner.affiliate.com` domain.

Using the Visual IVR URL

You can test and customize the link at the bottom of the DNIS tab of the campaign properties. This link enables you to create a Visual IVR session, which opens a Visual IVR flow in your browser. You can test the flow of any script associated with the campaign. The URL is generated specifically for each campaign. The link is in this format:

```
<base_url>/<domain_id>/campaigns/<campaign_id>/new_ivr_session  
base_url: http://api.five9.com/ivr/<Visual IVR API version>  
domain_id: ID of your domain  
campaign_id: ID of your campaign
```

Example

```
http://api.five9.com/ivr/1/12345/campaigns/123/new_ivr_session
```

Terminating the Visual IVR

At the end of the Visual IVR flow, you may add the `CALLBACK_URL` parameter if you want your customers to see a specific page. The optional parameter does not refer to callbacks or callback reminders but to any of the possible available methods to contact the customer, such as a specific Web page at your site, an email form, or a chat session.

Example

```
http://api.five9.com/ivr/1/12345/campaigns/123/new_ivr_session?__  
CALLBACK_URL__=http://www.myEmailForm.com
```

Integrating a Visual IVR Script in Your Web Site

To use a Visual IVR script, you need to customize the script to start Visual IVR and a button or link to a Web page so that the customer can access the Visual IVR. Start by copying the code from these files:

[IVRLaunch.js](#)
[vivrContainer.html](#)

Customizing the Script

Follow these steps to add the script to your Web site and to customize it.

- 1 Include the IVRLaunch.js script in your Web page by using a script tag.

```
<script src="/<PathToFile>/IVRLaunch.js"
type="text/javascript"></script>
```

- 2 Customize the file with your domain, campaign, and `messageError` properties:

```
var IVRLaunch = {
    host: "api.five9.com",
    version: "ivr",
    domain: "<My_Domain>",
    campaign: "<My_VIVR_Campaign>",
    layout: null,
    messageError: "<div style='height: 49px; display: table-cell;
vertical-align: middle; font-weight: bold; font-size:
30px;'><Please call us at +1 888.888.8888></div>"
};
```

- 3 At the bottom of the file, locate `vivrContainer.html` in the method named `IVRLaunch.init()`.

- 4 The script uses `vivrContainer.html` to present the VIVR content to your customers. You can modify the CSS as needed.

- 5 Add the path to the file in your Web site:

```
/** Load the Visual IVR container */
```

```
$.get("/<Path_To_File>/vivrContainer.html" + "?seconds=" +  
    (new Date()).getMilliseconds(),  
    function(data) {IVRLaunch.layout = data;}  
);
```

Adding a Button or Link to a Web Page

Place a button or link on your Web site to check if Visual IVR is available and, if so, to start a session. The script looks for `checkVIVR`, which is case sensitive. For example, this button uses Bootstrap CSS classes:

```
<a id="checkVIVR" class="btn btn-large btn-primary">Contact Customer  
Support</a>
```

Customizing the Visual IVR Page

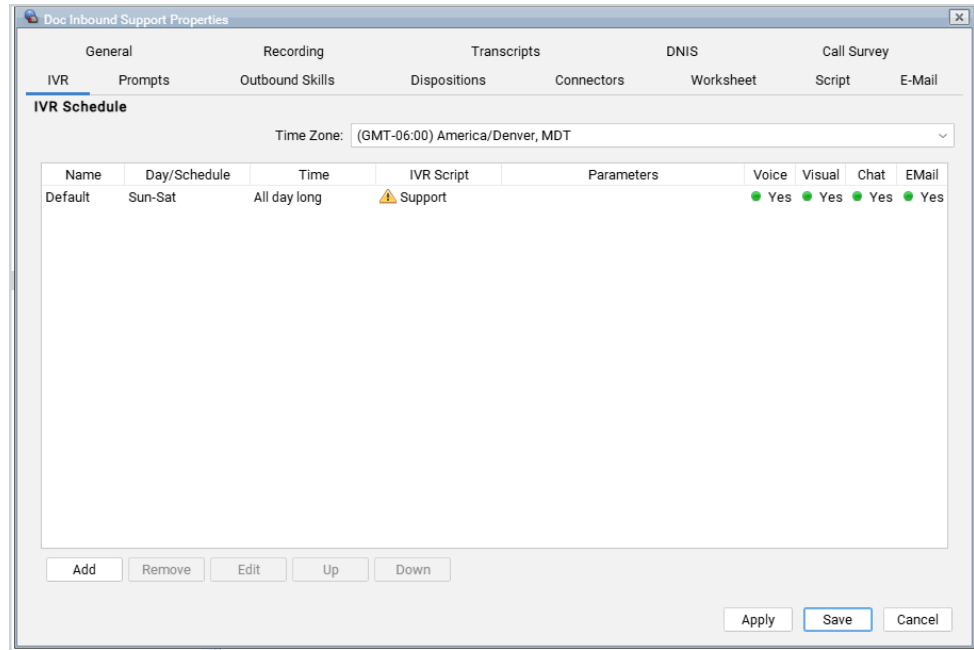
Follow these steps to customize the presentation of your Visual IVR page:

Note

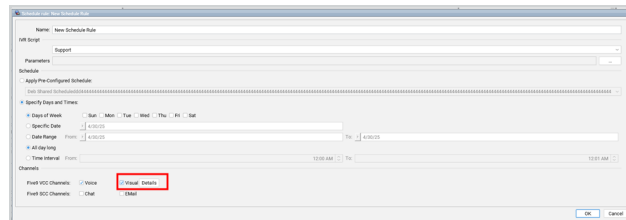
You must be a developer to modify the CSS.

- 1 In VCC administrator application, open the campaign properties for the VIVR campaign that you want to modify.

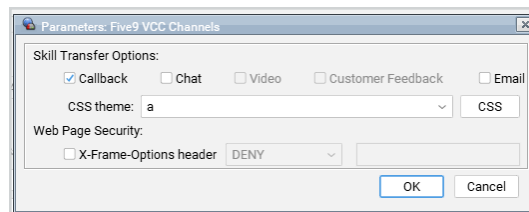
- 2 Click the **IVR** tab, then double-click the schedule rule.



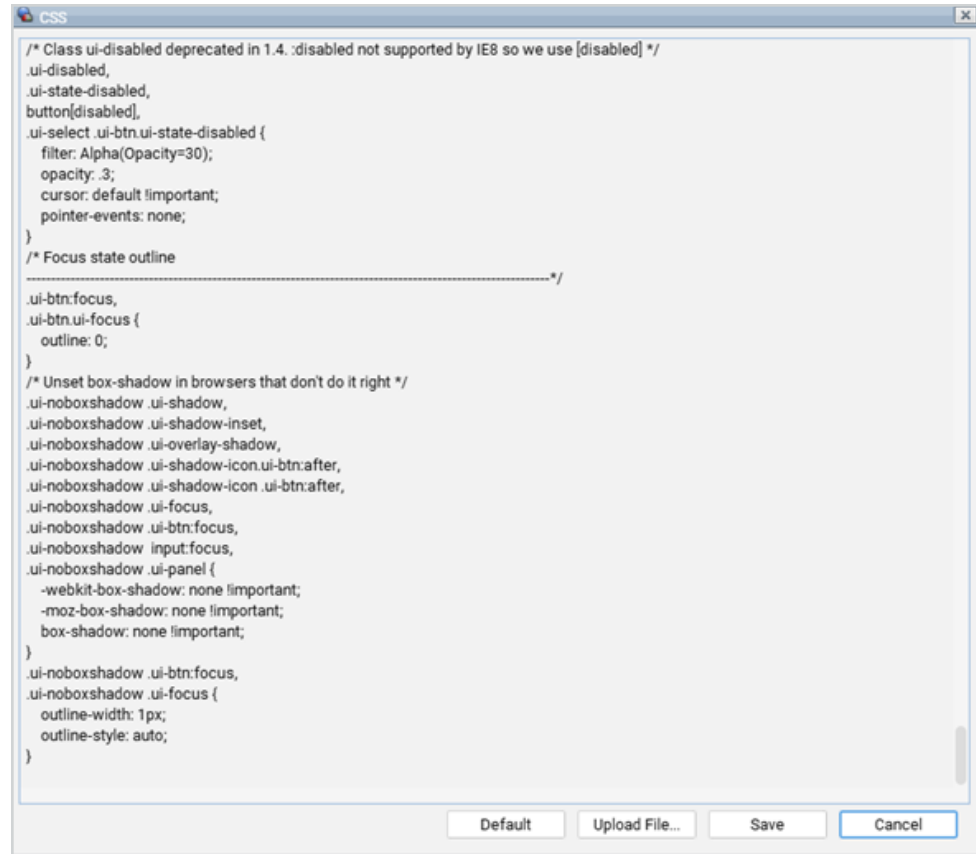
- 3 In the **Channels** section, click **Details** next to **Visual**.



- 4 Click **CSS** next to the **CSS theme** drop-down list, and edit the CSS as needed.



You may use the default settings or upload a file.



5 When done, click **Save**, and **OK**.

Preparing Visual IVR Scripts

You use IVR scripts for voice, visual, or both types of flows that contain voice and visual prompts. You can create specific Visual IVR scripts and convert existing voice- and digit-based scripts to Visual IVR scripts.

Choosing Modules

With Visual IVR flows, you can use most modules, and you need to understand the differences between voice and visual flows.

Available Modules

This table contains the modules that can be used in Visual IVR scripts.

Customer Interaction ¹	General IVR Flow ²
• Get Digits • Input • Lookup Contact • Menu • Play • Query • System Info • Voice Input	• Case • Contact Update • Foreign script • Hang up • If/else • Iterator • Set DNC • Set Variable • Skill Transfer • Update system info

¹These modules contain a Page tab where you can add prompts that are rendered visually in HTML.

²These modules can be connected and branched, and obtain or update information.

Modules with Collapsible Prompts

A few modules have an option named Collapsible that is used to concatenate all the prompts in the module whether they are played (voice) or displayed (Visual IVR).

When this option is enabled, callers see at once all the prompts in the same window. When this option is disabled, callers hear or see each prompt after performing the requested action.

You may add these prompts in the Prompts and Page tabs of these modules:

- Input
- Play
- Menu
- Voice Input

Modules that Accept Images

Full-size and thumbnail images added to Visual IVR scripts can improve your branding or help to describe a service or product. You can add images to these modules:

- Get Digits
- Input
- Lookup Contact Record
- Menu
- Play
- Query
- System Info
- Voice Input

You may add image files to the Page tab by specifying a URL or by adding the images as variables. When using variables, you can add the image to any variable group, and the name of the variable is arbitrary. The value of the variable must be the valid URL of an image. However, you cannot add links to images.

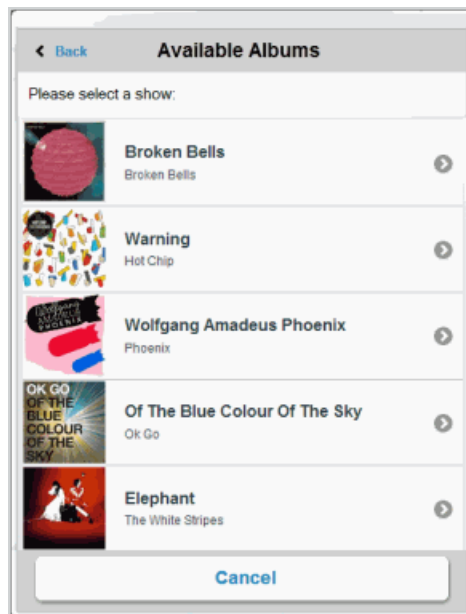
Example

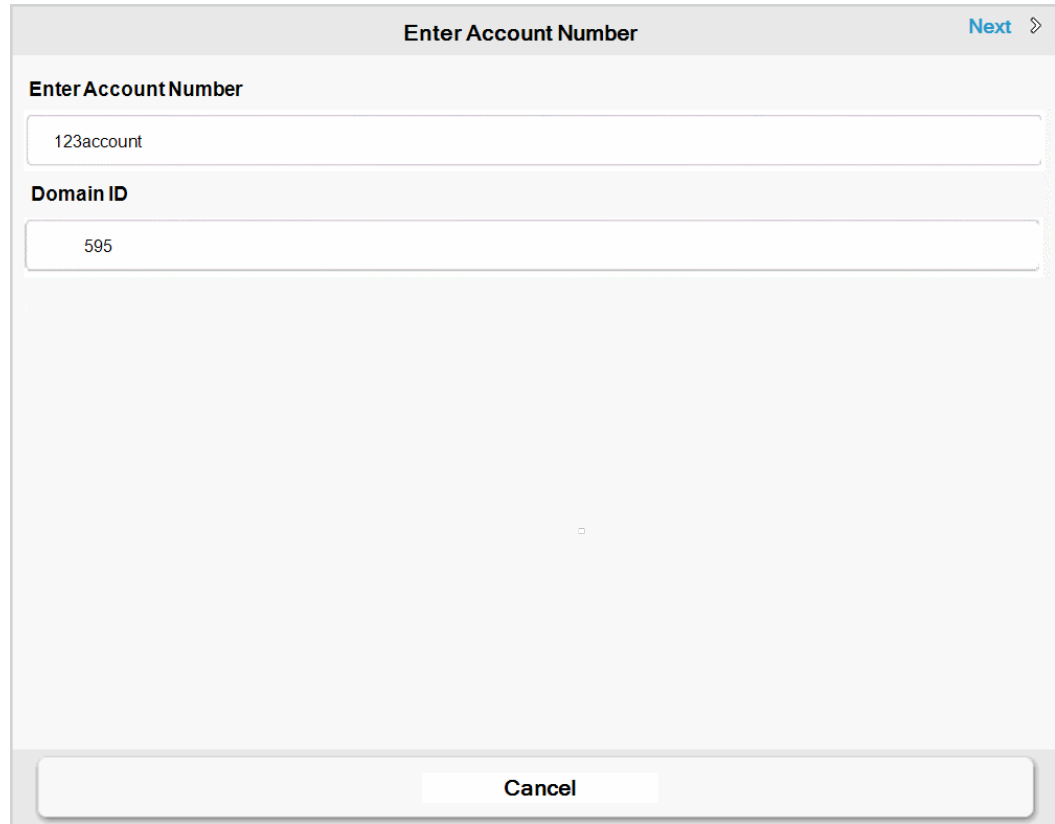
You may use this functional URL to test your variable in a Visual IVR script:

```
var = MyCav.str  
value = http://www.joelonsoftware.com/pictures/unicode/ascii.png
```

Example

This menu enables the customer to see a preview of an event, which facilitates the customer's decision. Each image is added to the Menu module.





The screenshot shows a window titled "Enter Account Number" with a "Next" button and a chevron icon in the top right corner. Inside the window, there is a section labeled "Enter Account Number" with a text input field containing "123account". Below this is a section labeled "Domain ID" with a text input field containing "595". At the bottom of the window is a "Cancel" button.

- 1 On the **General** tab, enter this information:
 - Module Name: Enter the text that you want your callers to see in the heading of the IVR window, for example: *Enter account Number*.
 - Collapsible: Enable this option so that both prompts appear to callers the

same HTML page.

The **Input Module** can be used to analyze spoken caller input or DTMF for short phrases based on pre-defined grammars for numbers, credit cards, zip codes, social security numbers, phone numbers, dates, time, and Boolean (Yes/No) phrases, so that the call will follow the appropriate branch in the IVR application.

Module Name:

Variable:

Grammar:

Properties:

Appl	Property	Value
☒	language	en-US
☒	y	1
☒	n	2

Additional Returned Values:

Return Key	Variable to Save Value
SWL_literal	<Ignore>
Confidence Level	<Ignore>

Max Attempts:

DTMF Help Button:

☐ Record User's Input as Audio File

☒ Collapsible

- 2 On the **Page** tab, add and configure as HTML the options that appear to callers on the same HTML page in the Visual IVR script.

Both prompts appear in the list of prompts.

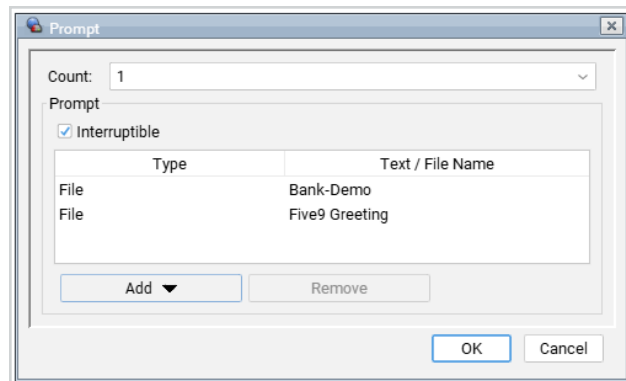
Header:

Type	Text / File Name
Text	Test Your Account Number
Text	Test Domain ID

Body:

Type	Text / File Name
------	------------------

- 3 On the **Prompts** tab, add voice prompts that are similar to the visual prompts for callers who are not using a mobile device.

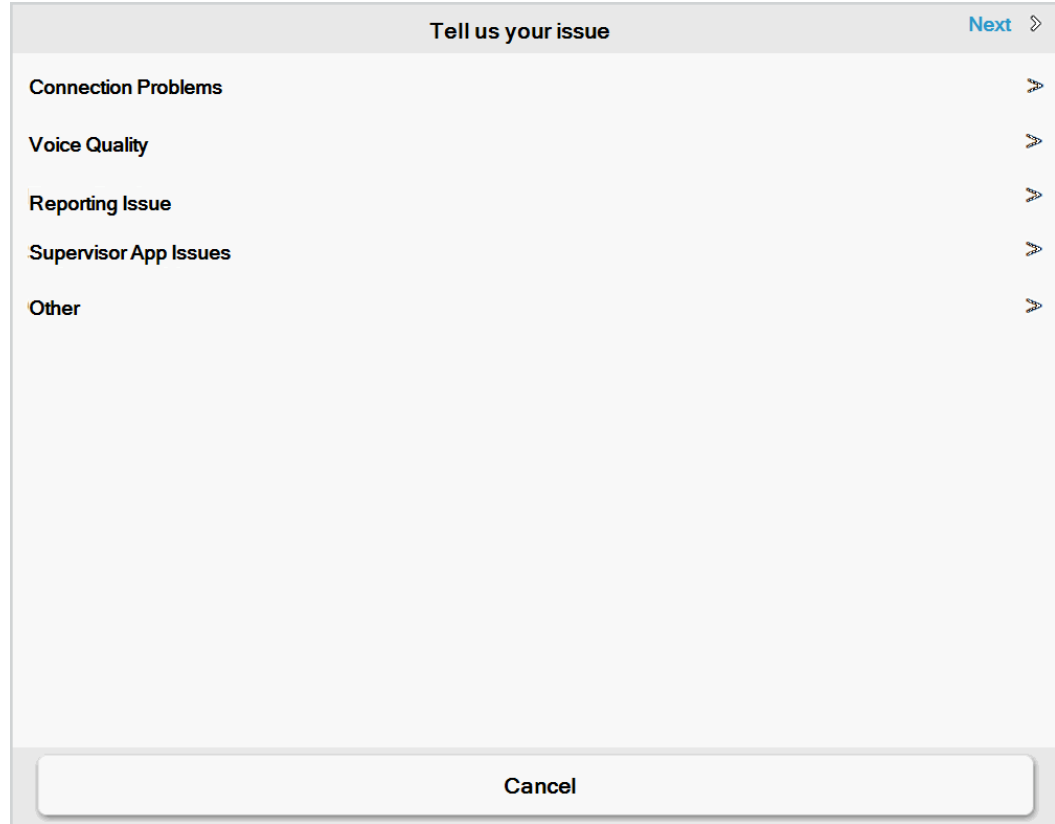


Lookup Contact Record Module

This module is used to search the database for the callers' information to present the most appropriate prompts. Use the most appropriate search method, such as **Use Call Attached Contact Record**. This module is not visible as HTML prompts to callers.

First Menu Module

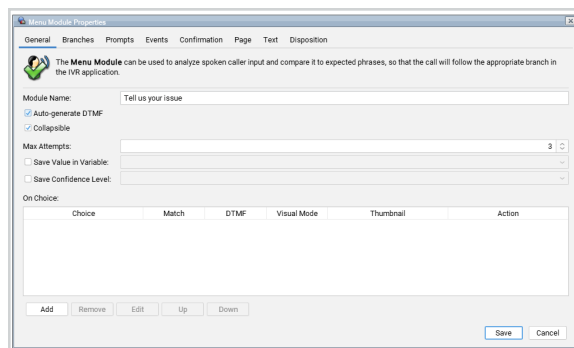
This module is used to list the options available to callers. Each option is a branch in the Visual IVR script. If you have more than one level of options, you can link multiple **Menu** modules. For more information, see the [Menu Module](#).



The screenshot shows a web-based IVR menu titled "Tell us your issue". In the top right corner, there is a "Next" button with a right-pointing chevron. The main content area lists five menu items, each with a right-pointing chevron icon to its right: "Connection Problems", "Voice Quality", "Reporting Issue", "Supervisor App Issues", and "Other". At the bottom of the interface is a large, light gray "Cancel" button.

1 On the **General** tab, enter this information:

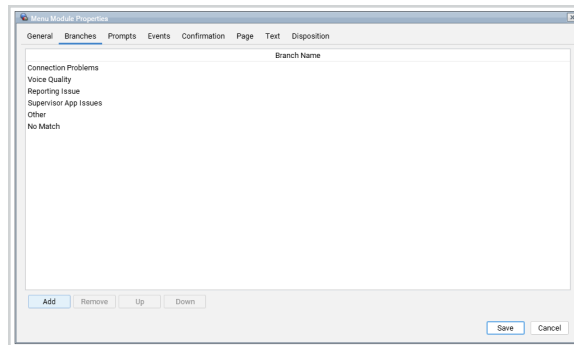
- Module Name: Enter the text that you want your callers to see in the heading of the IVR window, for example: *Tell us your issue*.
- Collapsible: Enable this option so that all choices appear to callers on the same HTML page.



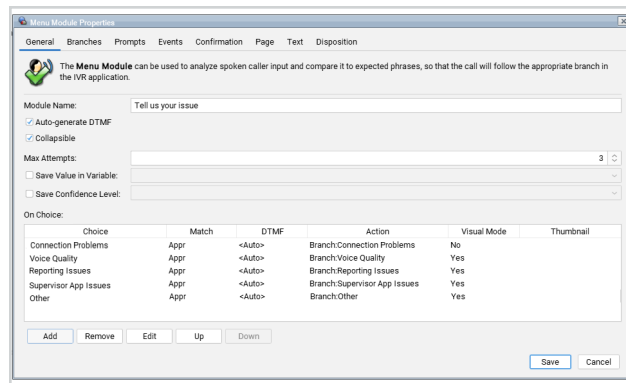
The screenshot shows the "Menu Module Properties" dialog box with the "General" tab selected. The "Module Name" field contains "Tell us your issue". The "Auto-generate DTMF" and "Collapsible" checkboxes are checked. The "Max Attempts" field is set to 3. Below these are unchecked checkboxes for "Save Value in Variable" and "Save Confidence Level". The "On Choice:" section contains a table with columns: Choice, Match, DTMF, Visual Mode, Thumbnail, and Action. At the bottom of the table are buttons for "Add", "Remove", "Edit", "Up", and "Down". At the bottom right of the dialog are "Save" and "Cancel" buttons.

2 On the **Branches** tab, create a branch for each choice, in this case:

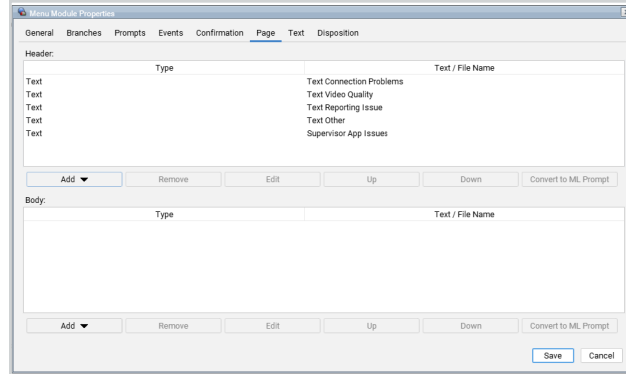
- Connection Problems
- Voice Quality
- Reporting Issues
- Supervisor App Issues
- Other



3 On the **General** tab, add an exit option for each branch.



4 On the **Page** tab, add and configure as HTML the options that appear to customers.



- 5 On the Prompts tab, add voice prompts that are similar to the visual prompts for callers who are not using a mobile device.

Skill Transfer Modules

This module is the only transfer module that you can use in Visual IVR scripts. You need to transfer the callers' requests to skill groups. Each branch is connected to a skill group. The configuration of each skill transfer module is not specific to Visual IVR scripts. This module is not visible as HTML prompts. For more information, see [Skill Transfer Module](#).

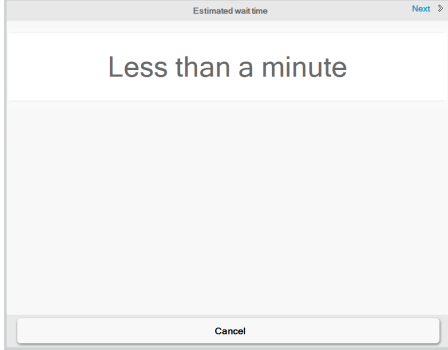
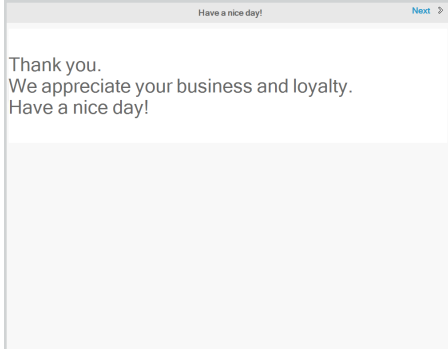
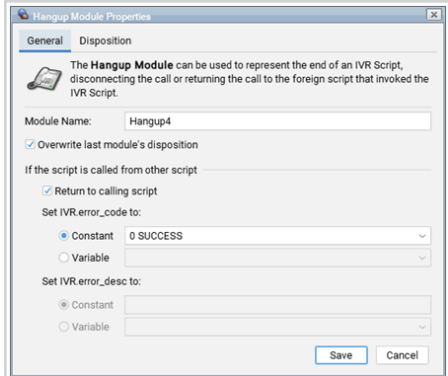
Second Menu Module

If the contact options, such as phone, chat, and email, are the same for all the possible issues, you can connect all Skill Transfer modules to a single Menu module with three branches.

Phone Branch

This branch contains four modules.

Visual IVR	Module
	Input Module This module enables the customer to enter a callback phone number.

Visual IVR	Module
	Play Module This module shows the estimated wait time obtained from a call variable such as <code>Call.hold_time</code> .
	Play Module This module contains a message in HTML format for callers. You could also add the message as HTML text below the estimated wait time and omitting this module.
	Hang Up Module When the customer closes the window, this module terminates the IVR session and assigns a disposition to it.

Chat Branch

This branch contains a Play module with an estimated wait time and a message, such as *Please wait for an agent*.

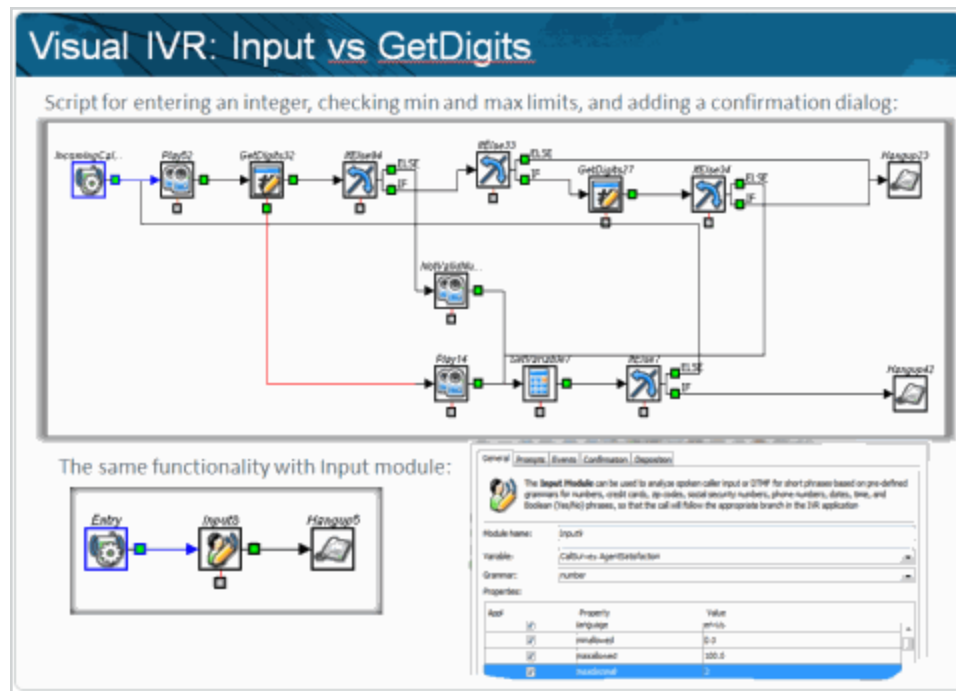
Email Branch

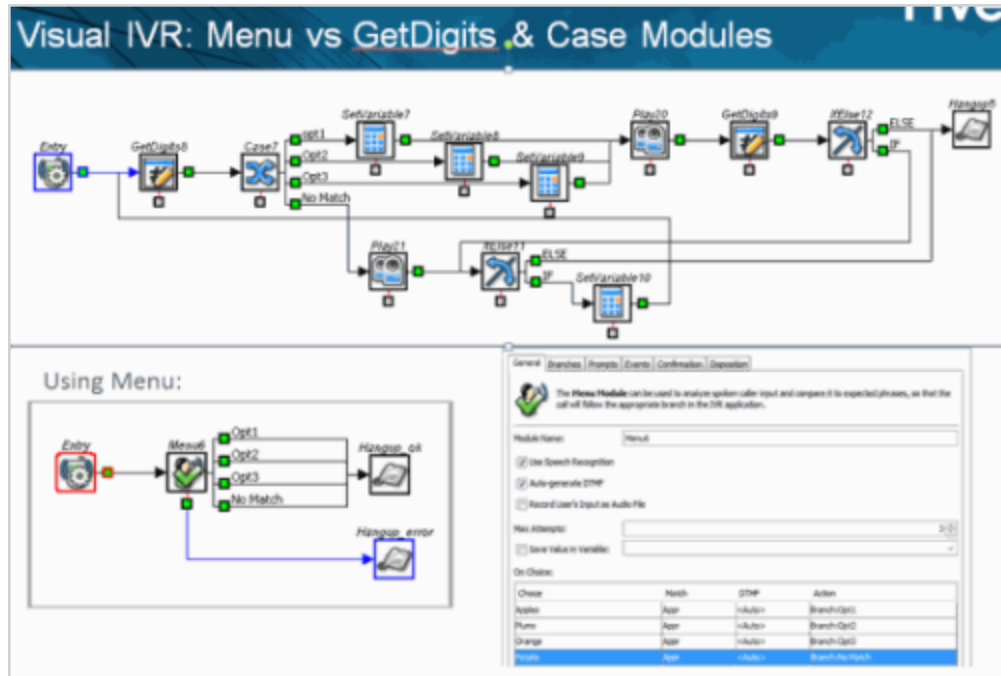
This branch is similar to the Phone branch. An Input module enables the customer to enter an email address. A Play module shows an HTML message with an estimated wait time. The Hang Up module ends the session and assigns a disposition.

Converting Voice- and Digit-Based Scripts into Visual IVR Scripts

By combining several of these modules in your existing IVR scripts, you can simplify your IVR scripts:

- Play
- Menu
- Get Digits
- Voice Input
- Input
- Skill Transfer





Five9 Engagement Workflow

Five9 Engagement Workflow enables you to harness the power of either or all of the [Five9 Rules Engine Module](#) and Natural Language Processing (NLP), [Query Module](#) JSON responses for parsing, or [SFDC Einstein Module](#) capabilities to enhance your email interaction routing. More information about [Five9 Engagement Workflow](#) can be found in the [Digital Engagement Administrator's Guide](#).

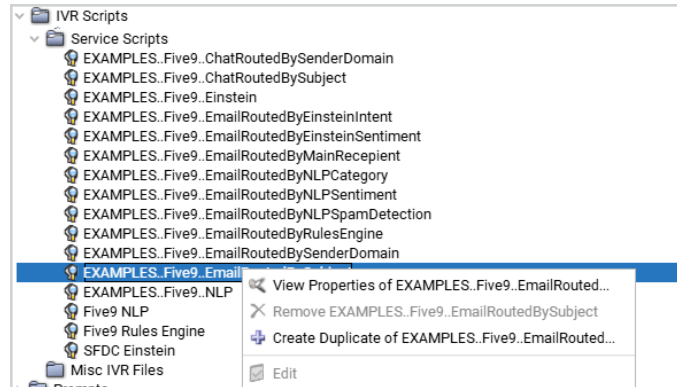
Five9 Engagement Workflow begins with your IVR script. In the simplest form, you can enhance email routing with such things as: call attached variables, contact profile information, disparate database lookups, or SFDC Einstein engine functions.

You can augment your routing decisions further by using NLP to determine such things as customer sentiment and urgency. Beyond this, you can filter spam from your distributed email interactions while recommending a next best action for relevant email interactions by scouring the email body.

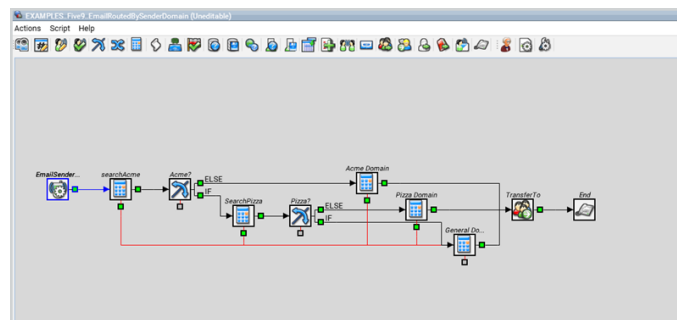
Configuring Engagement Workflow Templates

Five9 provides many templates to help with your IVR design. You can find these templates in the Five9 VCC Administrator application in the **IVR Scripts > Service Scripts** subfolder.

You can open example IVR scripts, as view only, or copy as a basis for your new script generation. Example IVR scripts are to be used for text (chat and email) interactions only.



Your copy of an example template enables you to easily populate your specific contact center details, such as data lookups, queues, and, in this example, email routed by subject.



To further categorize your engagement workflow routing and reporting, you can assign a custom disposition to your hangup module for engagement workflow interactions. You must have, or create, matching disposition values in your VCC and Text Channel Administrator (SCC) applications. Depending on your Unified Dispositions setting, the rule engine will use either your VCC or SCC disposition for these interactions. This enables your IVR script to recognize and assign the custom disposition value in the hangup module. For more information about creating and assigning disposition values, see the [Basic Configuration Administrator's Guide](#).

You can find more detailed information about Five9 Engagement Workflow, NLP, Five9 Rules, Unified Dispositions, or SFDC Einstein in the [Digital Engagement Administrator's Guide](#).

Implementing IVR Scripts

This section describes how to configure and use IVR scripts.

[Managing IVR Scripts](#)
[Managing Variables](#)
[Adding Functions](#)
[Adding Prompts](#)
[Selecting Dispositions](#)
[Using Recorded Files](#)
[Importing Scripts](#)

When creating your IVR scripts, be aware of these limits:

- Maximum number of times that a script may enter any module during a call = 1000.
- Unlimited number of sub-scripts may be contained in a script.
- Maximum number of times that a script may call another script = 100. Each time that a script calls another, the counter increases by 1. The system does not differentiate between the same or different scripts.
- Maximum execution time for each script = 1 hour.

Managing IVR Scripts

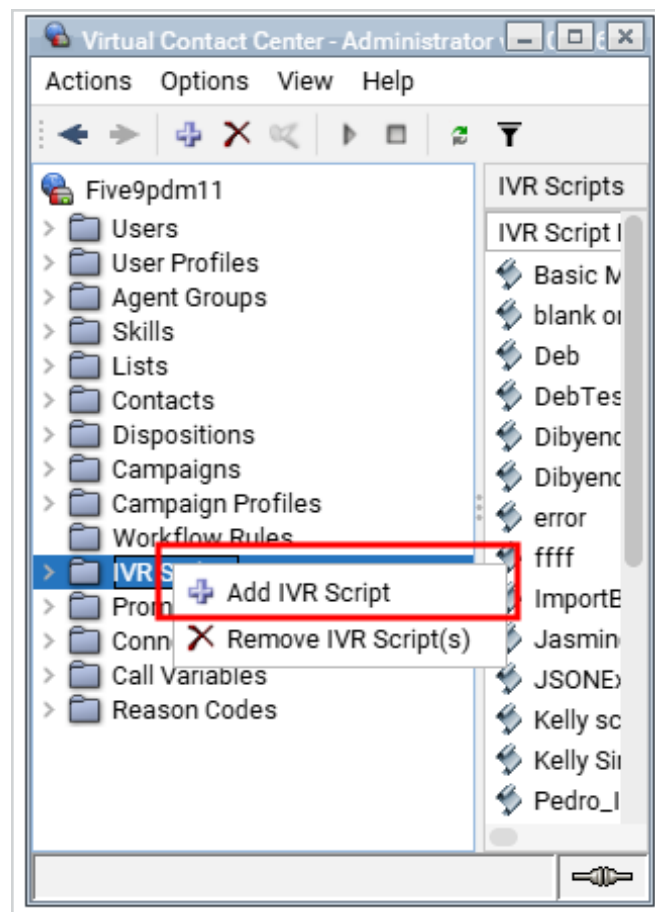
This section describes how to manage IVR scripts.

[Creating IVR Scripts](#)
[Duplicating IVR Scripts](#)
[Creating IVR Templates](#)
[Configuring IVR Modules](#)
[Updating IVR Scripts](#)
[Exporting and Importing IVR Scripts](#)
[Scheduling IVR Scripts in Campaign Properties](#)

Creating IVR Scripts

Before creating IVR scripts, plan your IVR flow, see the [Flowcharts](#), to determine which modules to use and how to connect and configure them. You will now be ready to create the flow for a test campaign. After testing, use the export function and add it to your production campaign.

- 1 In the Navigation pane, right-click IVR Scripts, and select **Add IVR Script**.



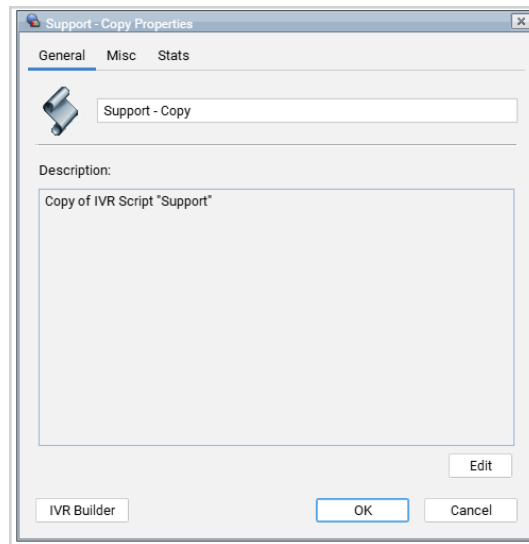
- 2 Enter the name of the IVR Script.
- 3 Click **OK**.

Duplicating IVR Scripts

If you duplicate the IVR Scripts, all modules and their relationships are copied.

- 1 In the navigation pane, right-click a script, and select **Create Duplicate of <name>**.

All properties are copied to a new object called **<original object name> - Copy**. The **Property** window of the duplicate object opens.



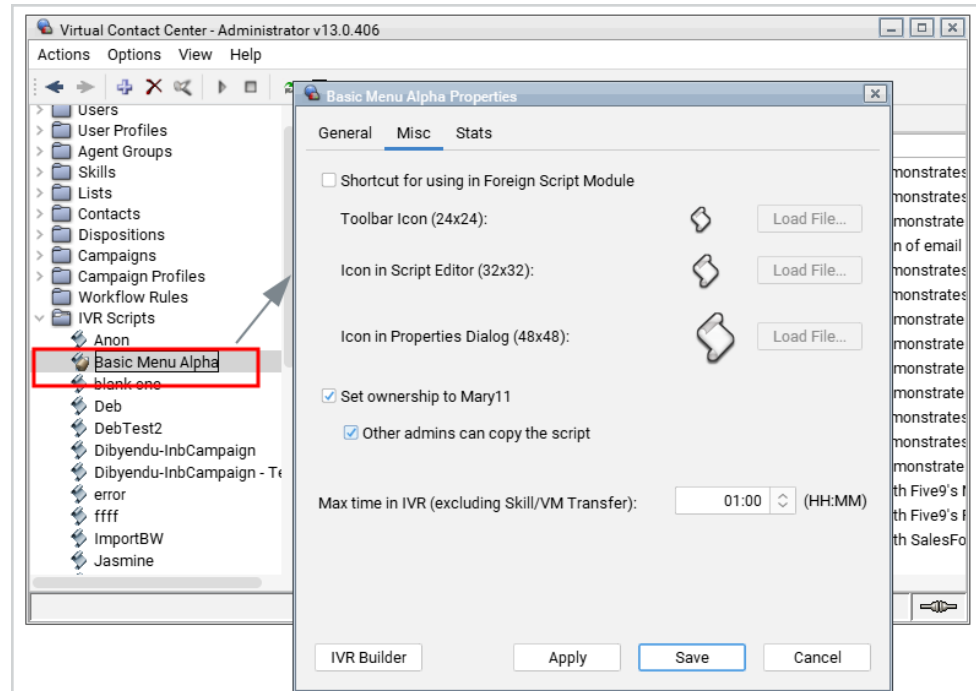
- 2 Fill in the required fields.

Creating IVR Templates

You can create a template from any IVR script to more efficiently copy and share repeatable processes or procedures. Template ownership is assigned to the administrator creating the template. You can choose to allow other administrators to copy the script or permit only yourself this privilege. Once the script has been assigned ownership it is stored in the Service Scripts folder of IVR Scripts to be identified among the other reusable scripts in your VCC.

To create an IVR template, follow these steps.

- 1 Right-click or double-click to access the properties of the IVR script from which this template will be made.



2 In the **Misc** tab, select icons to associate with this IVR template.

Icons must be PNG files. You must set the following three display icons:

- Toolbar icon: 24x24 pixels.
- Icon in IVR builder: 32x32 pixels.
- Icon in properties dialog: 48x48 pixels.

3 To convert this script into a reusable template, click **Set ownership to <your user name>**.

To make this template available to other administrators, enable **Other admins can copy the script**.

4 Click **Save**.

This IVR script is now a reusable IVR template stored in the **Service Scripts** subfolder of **IVR Scripts**.

Configuring IVR Modules

Most IVR modules must be configured. Double-click an icon, or right-click the icon and select **Module Properties**.

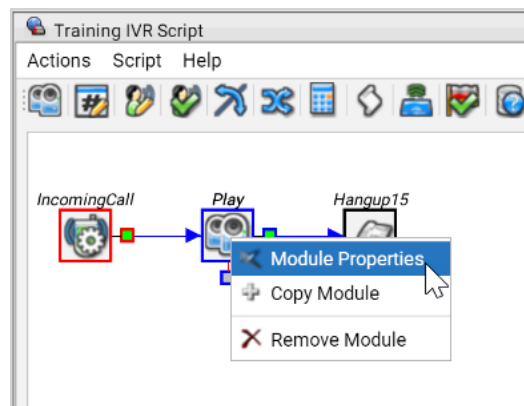
[Adding and Removing Modules](#)

[Selecting Icons](#)

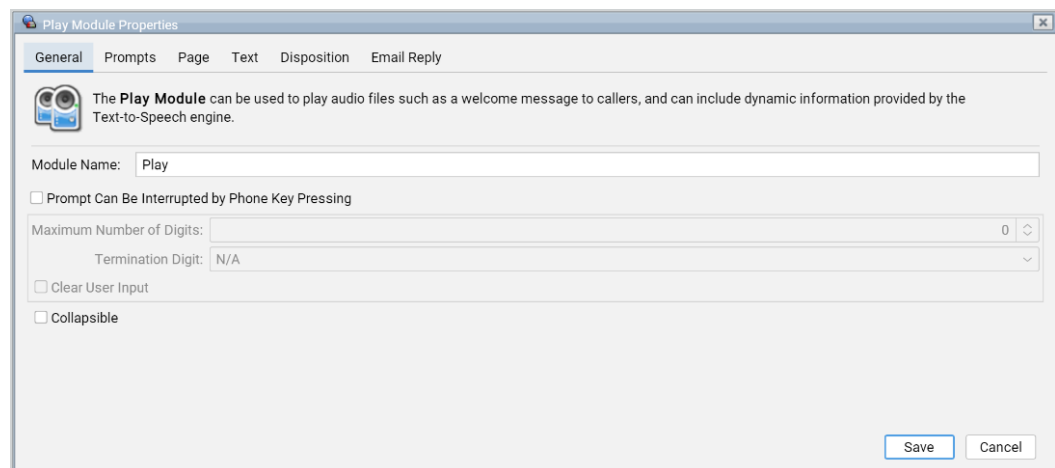
[Connecting Icons](#)

[Removing Connections](#)

[Clearing the Work Area](#)



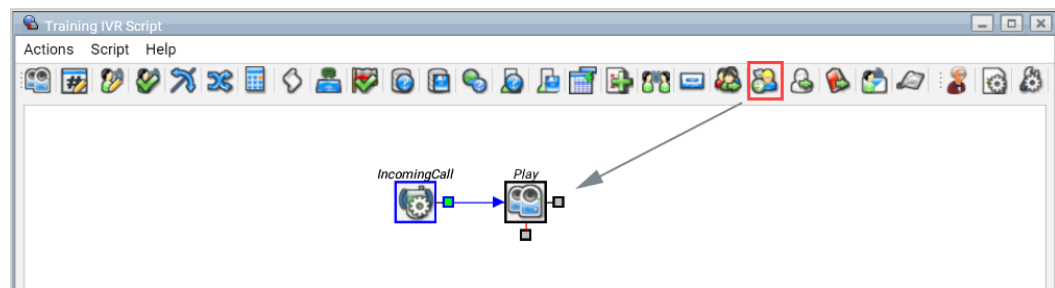
The **Module Properties** window may have one or more tabs. Define the necessary options in the tabs.



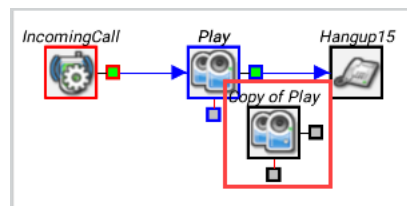
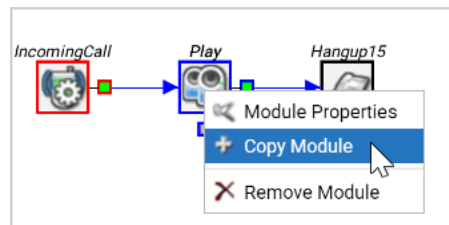
Name each IVR module when creating the script. The IVR module name is recorded in the IVR_PATH field in reports allowing you to view the IVR path for every call. For more information, see [IVR Modules](#).

Adding and Removing Modules

IVR modules are represented by icons in the IVR toolbar. Place your cursor on an icon to view its name. You can drag and drop the icons from the toolbar to the workspace. The Incoming Call icon is in the workspace by default.



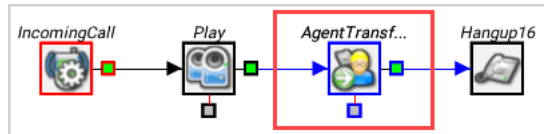
You can copy icons with their properties by right-clicking an icon and selecting **Copy Module**.



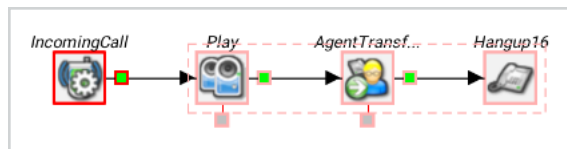
The new module is displayed in the working area. To remove a module, right-click the icon and select **Remove Module**, or press <DELETE>.

Selecting Icons

Click an icon. The selected icon has a blue border, and the connected modules are indicated by the blue connector lines.

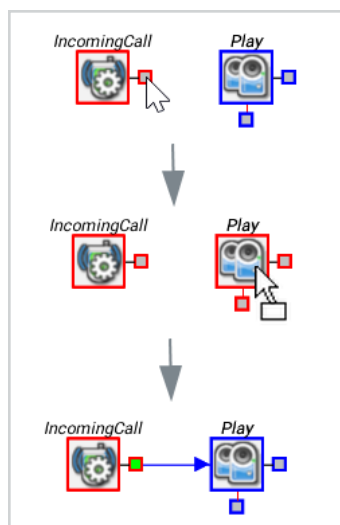


To select more than one module, click and drag over an area to include selected modules. You see a dotted box that indicates the selection. Once selected, these modules can be dragged to another location or deleted as a group.



Connecting Icons

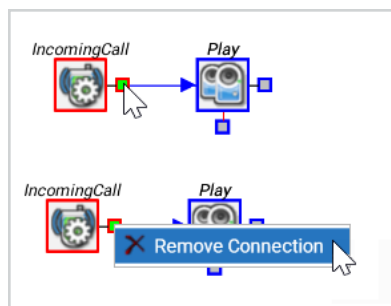
Each module has output ports so you can drag and drop some of them to the destination module. To connect one module to another, drag and drop the connector box to the following module icon. The connected module icons are now linked by an arrow:



Some IVR modules provide branching. For example, If/Else and Case. Each IVR branch must be created and terminated individually. You cannot link from one branch into another branch. You can form loops by connecting a module to a previous module. See also [Creating an IVR Loop](#).

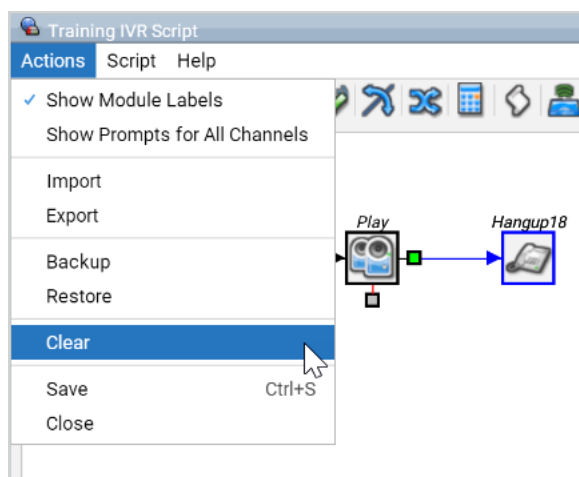
Removing Connections

To remove a link, right-click the connection box and select **Remove Connection**.



Clearing the Work Area

To clear the working area, select **Actions > Clear** in the menu. This action removes all modules except Incoming Call.



Updating IVR Scripts

You need to configure new IVR scripts and those with changed properties. If changes are saved while the associated campaigns are running, new calls use the updated script.

- 1 Open the IVR Scripts folder.
- 2 Right-click an IVR script, and select **Edit**, or double-click the IVR Script.
In the IVR Script Designer, you can create a script, configure each module, define optional variables and properties, and import a script.
- 3 When done, select **Actions > Save**.

When saving a script, the system checks the IVR flow and displays an error message if a problem is found. All errors must be fixed before the script can be saved.

Exporting and Importing IVR Scripts

[Exporting IVR Scripts](#)
[Importing IVR Scripts](#)
[Backing Up IVR Scripts](#)
[Restoring IVR Scripts](#)

The IVR Script Designer includes options to export and import your IVR Scripts. Use this feature to make periodic backups of your IVR Scripts and to save IVR Scripts that are in progress. This feature can be utilized by call centers running multiple campaigns and or with multiple clients to import or export IVR scripts between campaigns.

You can export and import a complete or incomplete script. This function works with the flow chart, variables, and default properties. You can call this function from the Actions menu of the IVR Script Designer.

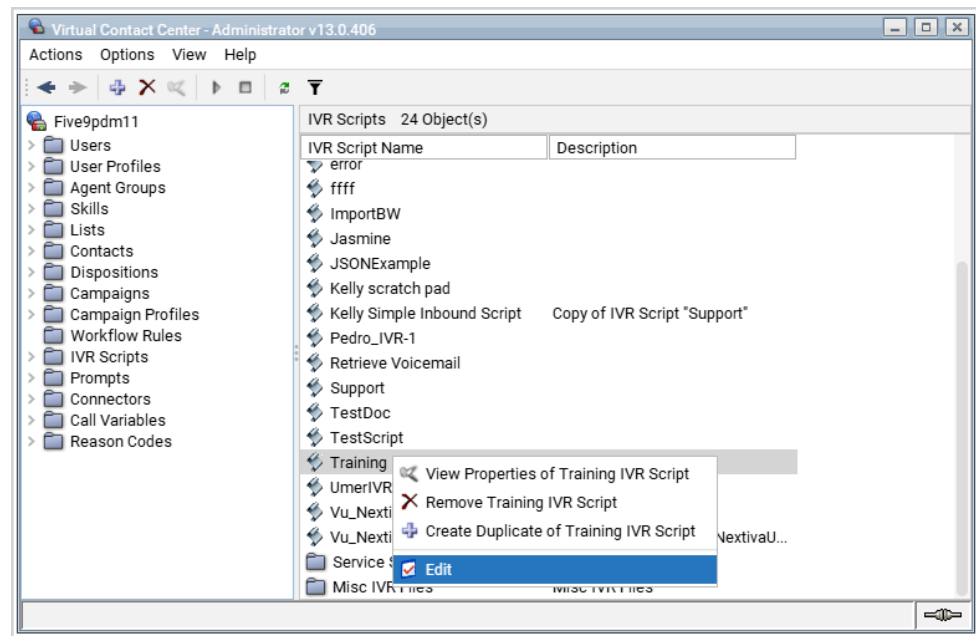
Autodial IVRs are not compatible with standard inbound campaign IVRs. If you are using a standard inbound campaigns, you cannot use autodial IVRs in an inbound campaign and you cannot use inbound IVRs in autodial.

Note: If you are using VCC Administrator Console as a Web Application, additional steps are required to save, import, export, and backup. See [File Manipulation in Java Virtualization](#).

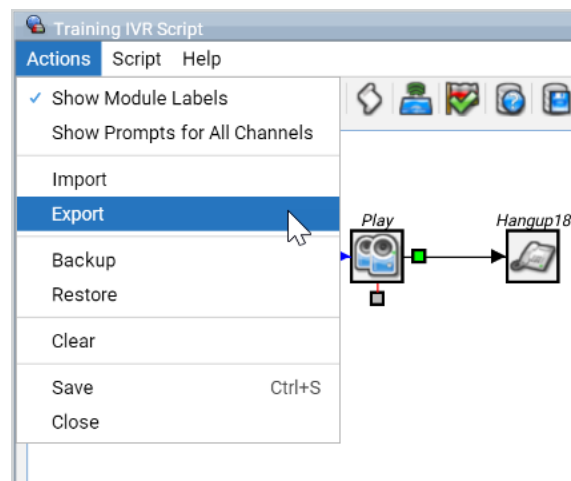
Exporting IVR Scripts

Follow these steps.

- 1 In the navigation pane, select **IVR Scripts**.
- 2 Right-click a script name and select **Edit**, or double-click the IVR Script.



- 3 Select **Actions > Export**.



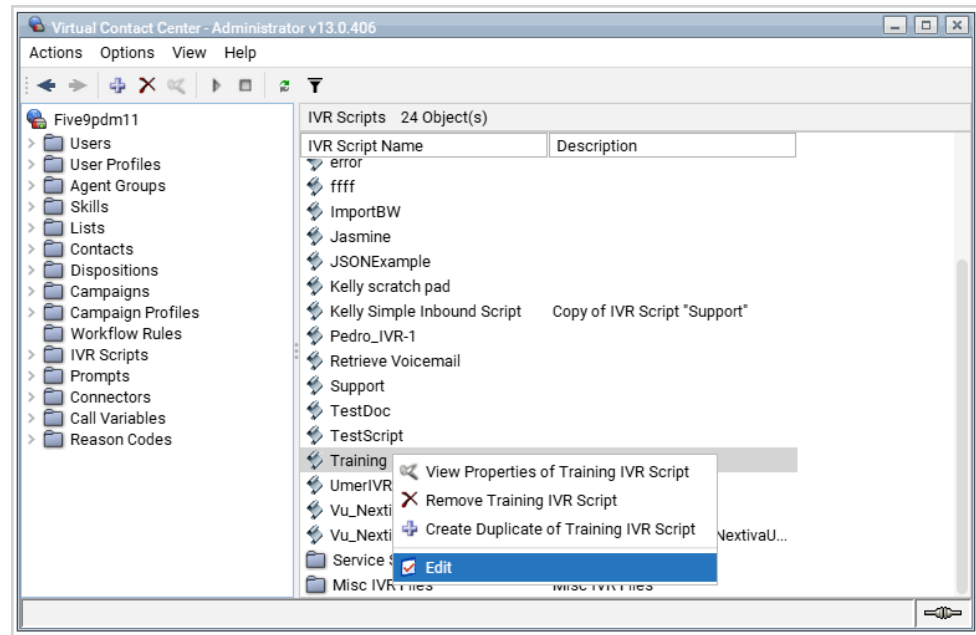
4 Save the file on your computer.

The file extension is `five9ivr`. If you try to save an unfinished IVR Script, you receive an error message. By clicking Export Script, you can export the call as a separate file on your computer.

Importing IVR Scripts

Follow these steps. The imported data overwrites the existing data.

- 1 In the navigation pane, select **IVR Scripts**.
- 2 Right-click a script name and select **Edit**, or double-click the IVR Script.

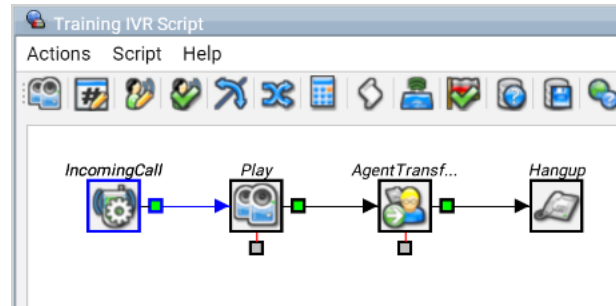


3 Select **Actions > Import**.

4 Locate a file with extension `five9ivr`.

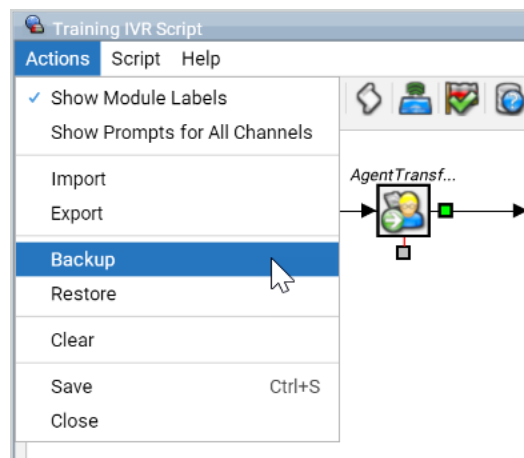
5 Click **Open**.

The script is displayed in the working area.



Backing Up IVR Scripts

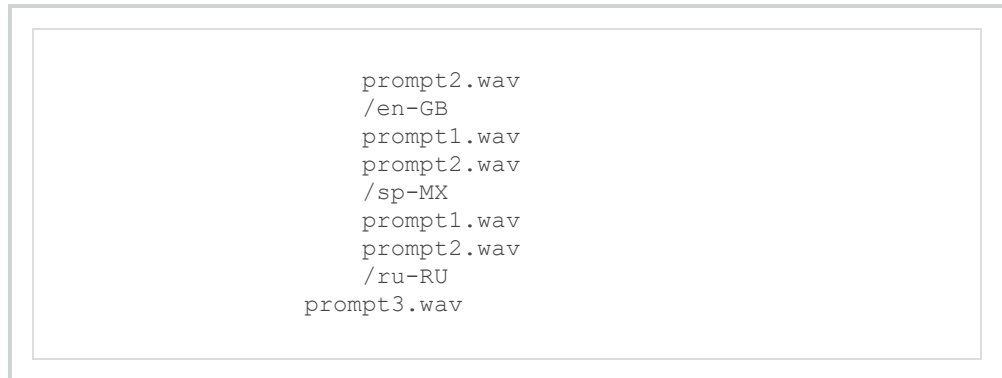
You can export the IVR script, including all subscripsts and users prompts. System prompts aren't included in the archive. This process applies to all supported languages. To create a backup of an IVR script, select **Actions > Backup** in the IVR script builder.



The following example shows the exported elements of an existing IVR script zip file.

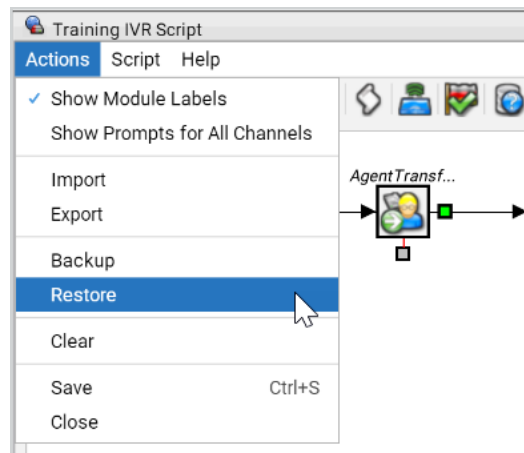
Example

```
<name = ScriptName>.zip --->
    description.txt
    description.xml
    /SCRIPTS
    IVRname.five9ivr
    subscript_1.five9ivr
    /PROMPTS
    prompt1.wav
```



Restoring IVR Scripts

To restore an IVR script with its subscripsts and prompts, navigate to **Actions > Restore** from in the IVR script builder and select the appropriate zip file to restore.



Scheduling IVR Scripts in Campaign Properties

You can schedule an IVR script in the the IVR tab of the campaign properties in inbound and autodial campaigns. The IVR tab enables you to schedule the scripts by days of week, holidays, time ranges, and date ranges.

- 1 In the **Navigation** pane, right-click or double-click on a campaign to view the properties window.

The **Inbound Properties** window displays. The default script is always enabled. Additional scripts override the default script.

2 Select the **IVR** tab.

The screenshot shows the 'Inbound Properties' dialog box with the 'IVR' tab selected. The 'IVR Schedule' section is visible, showing a table with one entry: 'Default' with 'Sun-Sat' and 'All day long'. The 'Time Zone' is set to '(GMT-07:00) America/Los_Angeles, PDT'. The table has columns for Name, Day/Schedule, Time, IVR Script, Parameters, Voice, Visual, Chat, and EMail. The 'Voice' column shows 'Yes' with a green checkmark. The 'Visual' column shows 'Yes' with a green checkmark. The 'Chat' column shows 'Yes' with a green checkmark. The 'EMail' column shows 'Yes' with a green checkmark. Below the table are buttons for 'Add', 'Remove', 'Edit', 'Up', and 'Down'. At the bottom right are 'Apply', 'Save', and 'Cancel' buttons.

3 Select **Add** to add an IVR schedule.

The screenshot shows the 'Schedule Rule: New Schedule Rule' dialog box. The 'Name' field is set to 'New Schedule Rule'. The 'IVR Script' field is empty. The 'Parameters' field is empty. The 'Schedule' section is expanded, showing 'Apply the Configured Schedule'. Under 'Specify Days and Times', the 'Days of Week' section is selected, showing 'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', and 'Sat'. The 'Specific Date' section is also visible, showing 'From' and 'To' dates. The 'All day long' section is selected, showing 'From' and 'To' times. The 'Channels' section is expanded, showing 'Voice' and 'Visual' channels. The 'Voice' channel is selected, showing 'Details'. The 'Visual' channel is also visible, showing 'Details'. The 'Chat' and 'EMail' channels are also visible. At the bottom right are 'OK' and 'Cancel' buttons.

4 Enter a name for the new schedule rule.

5 Specify Days and Times.

- **Days of the Week** – Select a Day/Days when the IVR is applicable.
- **Specific Date** – Set a specific date to trigger the IVR schedule. In case of multiple dates, add different IVR schedules.

- **Date Range** – Set a date range when the schedule starts and stops.
- **All Day Long** – Enable for 24-hour schedule.
- **Time Interval** – Set a time when the IVR will use the schedule.

If the call is out of coverage of this schedule then it would follow the **Default Schedule** and its corresponding IVR script.

Name	Day/Schedule	Time	IVR Script	Parameters	Voice	Visual	Chat	E-Mail
Business	6/5/25 - 6/21/25	All day long	Five9 Rules En...		Yes	-	-	-
Default	Sun-Sat	All day long	Dibyendu-Inb...		Yes	Yes	Yes	Yes

Managing Variables

You may use two types of variables:

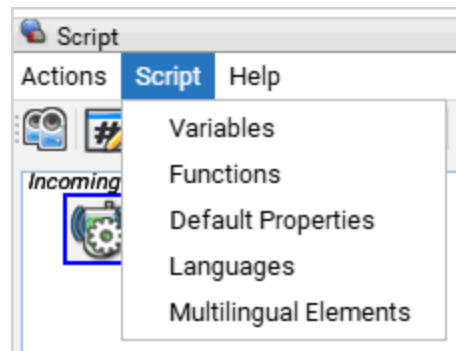
- Script variables that you create and use in each script.
- Call variables that you create and use in the VCC.

Script Variables

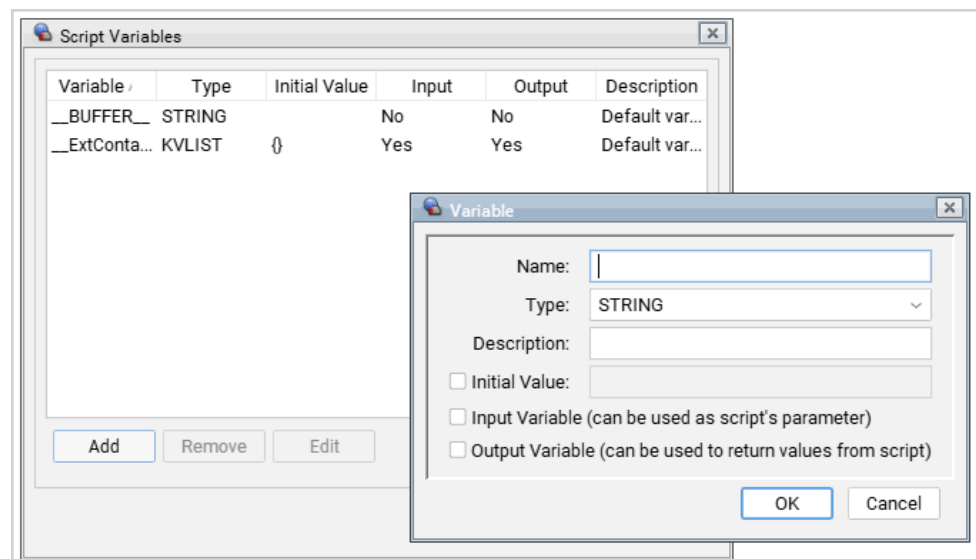
This section describes how to create and modify custom variables.

Adding Variables

- 1 In the IVR Script Designer, select **Script > Variables**



- 2 Click **Add**.
- 3 For each variable, specify these elements.



- **Name:** Strings with initial letter. Cannot contain spaces or start with two underscores (reserved for system variables).
- **Type:** Data type for the field:
 - **STRING:** Alphanumeric, maximum 250 characters>
 - **INTEGER:** Whole number

- **NUMERIC:** Number with decimal. Can be used for very large numbers.
 - **CURRENCY:** Dollar, Euro, or British Pound
 - **DATE:** y-m-d, d.m.y, m/d/y, ymd
 - **TIME:** HHMM or HHMMp
 - **KVList:** {key1:value2, key2: value2 ...} where key and value is "string" or 'string'
 - key must be enclosed in double quotes. If you omit them, they are added.
 - Strings value must be enclosed in double quotes. If you omit them, they are added. Single quotes are changed to double quotes.
 - Quotes are optional for numbers.
 - Spaces before or after the colon or the comma are removed.
 - **Description:** Your description is copied as a comment into generated scripts prepared for Web requests.
 - **Initial value:** Whether to assign a default value to the variable. Variables that are not initialized are NULL (empty cells in the Variables tab). Empty strings have two double quotes.

String values can be entered as text or text surrounded by quotes. Internally surrounding quotation marks should be omitted, but the Variable tab shows all string values in the quotes. A variable of CURRENCY data type can be entered with or without the dollar (\$) sign. DATE and TIME can be entered in an arbitrary format, which can be interpreted correctly.
- Basic String Functions:
 - **String Length** – returns the number of characters in the input string
 - **String Left** – return the characters of the specified length starting from the left most character of the string
 - **String Right** – return the characters of the specified length starting from the rightmost character of the string

- **External value** – Can be used as the script's parameter. If a variable is marked External, its initial value can be set from the script schedule or from a calling script (foreign script). If the external values are specified at the script start time, the initial values in the table are not taken into consideration. The updated values of external variables are available in the calling script after exiting from the called script by passing reference parameters.

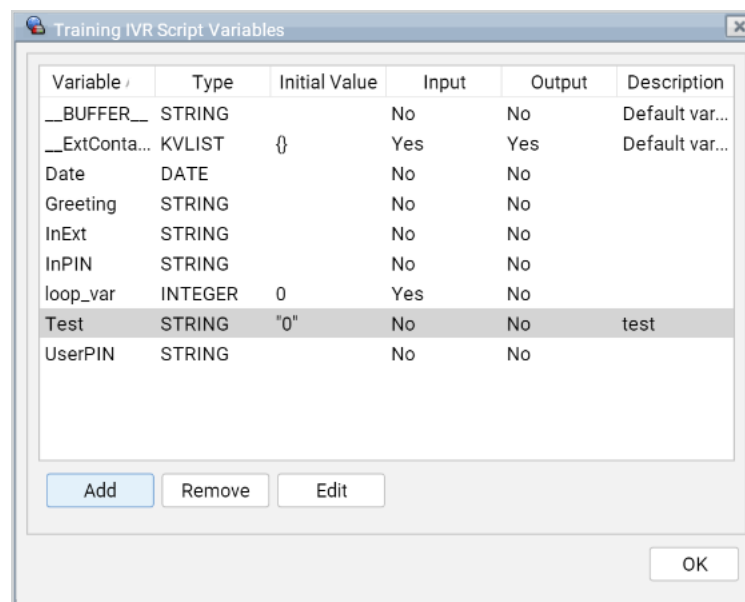
Note: The left and right functions only identify digits spaces, hyphens, and colons to split date/time components.

- 4 Click **OK**.

Editing and Deleting Variables

Before editing or removing variables, close all opened module properties menus.

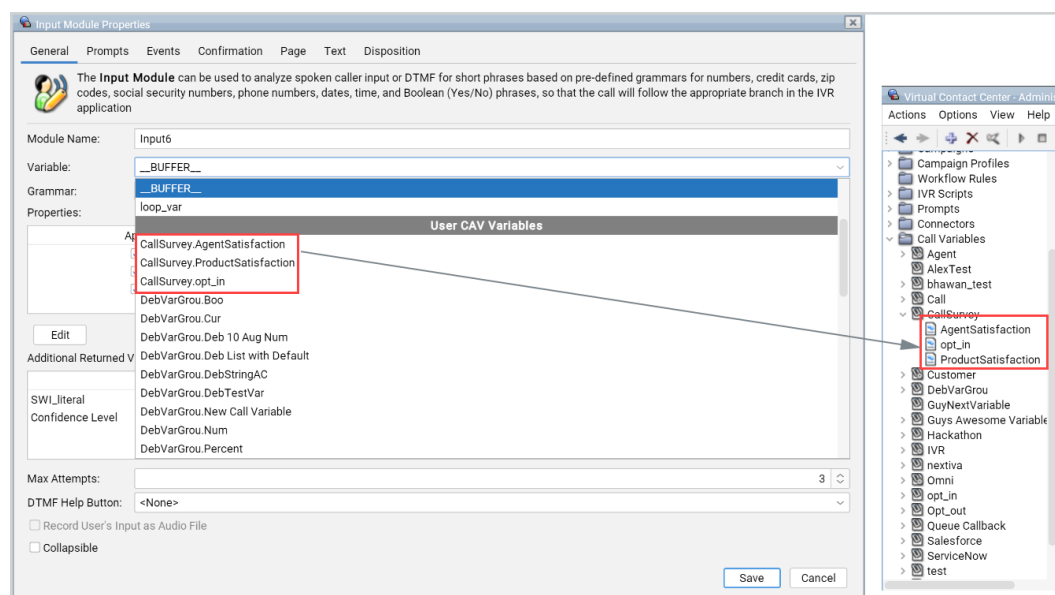
- 1 In the IVR Script Designer, select **Script > Variables**.
- 2 Double-click or select an item and select **Edit** or **Remove**.



- 3 If editing, make your changes, and click **OK**.

Call Variables

You can use call variables in IVR scripts. This figure shows the Input module properties window with the variables available for selection under the User CAV Variables title. In this case, these variables correspond to the group variable named VIVR in the Call Variables folder of the VCC. All IVR, system, and call variables support the UK and Euro currencies.



Although you may use variables in most modules, you may not use call variables that contain sensitive information in these modules:

- Agent/Voice Mail Transfer
- Foreign script
- Iterator
- Update system Info
- Voicemail transfer

For more information on managing and using variables, see [Variables](#).

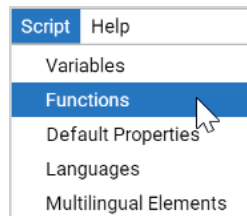
Adding Functions

You can apply JavaScript functions and arguments to parse the query response or to set variable values. You can add functions in these modules:

- Query:
 - You can define functions directly in the Query module **Response** tab, or select functions in the IVR builder **Script** menu.
 - Return values can be an array. Returned arrays can use a combination of Query and Iterator modules for greater flexibility.
- Set Variable:
 - You can use the Set Variable module with or without the Query module.
 - Returned arrays for the Set Variable module use only the first returned element. Other array elements are ignored.

To add a function, follow these steps.

- 1 In the IVR Script Designer, select **Script > Functions**.



- 2 Click **Add**.

Function

Name: Query1_parse

Type: KVLIST

Description: For parsing responses to Query 1

Function Arguments:

Type	Name	Description
KVLIST	body	Response for parsing respo...

Add Remove Edit Up Down

Function implementation (JavaScript):

```
function Query1_parse (body) {  
  
    function Five9 query1 Parse(body) {  
        var _ExtContactFields_array = [{}];  
        //....  
        return _ExtContactFields_array;  
    }  
  
}
```

Test

OK Cancel

3 For each variable, specify these elements.

- **Name:** Strings with initial letter. Cannot contain spaces or start with two underscores (reserved for system variables).
- **Description:** Description copied as a comment in generated scripts generated for Web requests.
- **Type**

- **STRING:** Alphanumeric, maximum 250 characters>
- **INTEGER:** Whole number
- **NUMERIC:** Number with decimal. Can be used for very large numbers.
- **CURRENCY:** Dollar, Euro, or British Pound
- **DATE:** y-m-d, d.m.y, m/d/y, ymd
- **TIME:** HHMM or HHMMp
- **KVList:** {key1:value2, key2: value2 ...} where key and value is "string" or 'string'
 - key must be enclosed in double quotes. If you omit them, they are added.
 - String value must be enclosed in double quotes. If you omit them, they are added. Single quotes are changed to double quotes.
 - Quotes are optional for numbers.
- **Function Arguments:** Define or select JavaScript functions. The Test button is enabled when all test values are specified.
- **Function Implementation:** Input values that contain the JavaScript function and corresponding arguments.

4 Click **OK**.

Adding Prompts

Several IVR modules have internally extended execution times. You can add prompts in these modules:

Module	Description
Agent/Voice Mail Transfer Module	A termination module that allows you to transfer calls to agents or to voicemail.
Conference Module	Enables you to add other parties to a conference call.

Module	Description
Get Digits Module	Enables you to route a call by using digits entered by the caller into a variable.
Input Module	Analyzes spoken words or DTMF input.
Lookup Contact Record Module	Enables you to retrieve information from the caller's contact record.
Menu Module	Enables you to configure multiple exit branches based on caller input.
Play Module	Enables you to play prompts or greetings anywhere in the script.
Query Module	Enables you to exchange database information with an external source.
Skill Transfer Module	Enables you to transfer calls to skill queues.
System Info Module	Enables you to retrieve properties and statistics.
Voice Input Module	Prompts the caller for information and saves the caller input.

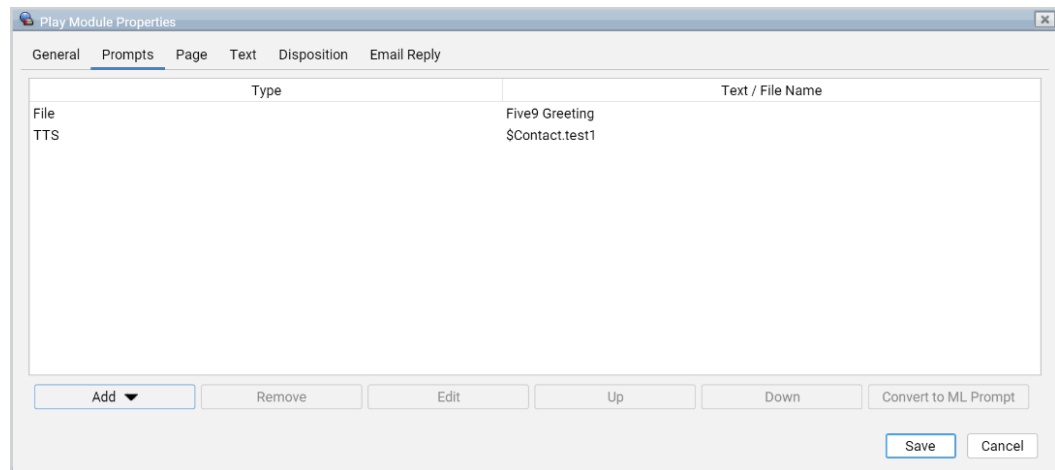
Adding Prompts

In the Prompts tab, click **Add** to select a prompt to be played by this module. The list of prompts includes the default prompts and the prompts that you uploaded in the Prompts folder. You can define one or several prompts to be played. You can indicate the sequence of prompts and how many times they should be used. Each prompt can be interruptible.

- Interruptible – Stops playing the prompt as soon as the background operation is completed, or the caller provides additional input.
- Exit the module in case of a prompt exception – indicates whether the IVR should move to the next module if there is an error in retrieving or playing a prompt.

TTS (enumerate) is a prompt element that opens a text-to-speech configuration window, with two additional variables: prompt and dtmf (if DTMF is enabled) available in Variable selection combo box. The Enumerate element is an automatically generated description of the available choices. It specifies a template that is applied to each choice in the order they appear in the menu. A default template is shown initially when

the Enumerate is added, but the user can change the text. If DTMF is enabled, the prompt generator prepares an appropriate prompt.



Selecting Dispositions

Dispositions are used only when calls are terminated or disconnected in these modules:

- [Extension Transfer Module](#)
- [Hangup Module](#)
- [Third Party Transfer Module](#)
- [Voicemail Transfer Module](#)

The **Dispositions** tab contains a default disposition that you can change to any system or custom disposition:

- Agent dispositions are used when a call is transferred to an agent.
- IVR dispositions are automatically used by the system when a call is disconnected during the IVR script. Any disposition, such as Send Email Notification and Add to DNC, can be assigned by the system. IVR dispositions enable you to see in reports where the caller was disconnected from the IVR script.

Using Recorded Files

Several IVR modules can collect audio information:

- [Conference Module](#)
- [Third Party Transfer Module](#)
- [Input Module](#)
- [Voice Input Module](#)
- [Menu Module](#)

The files can be processed by these modules:

- [Agent/Voice Mail Transfer Module](#)
- [Skill Transfer Module](#)
- [Hangup Module](#)
- [Voicemail Transfer Module](#)

These modules have the **Action for Recorded Files** menu. You can select these actions for the recorded files:

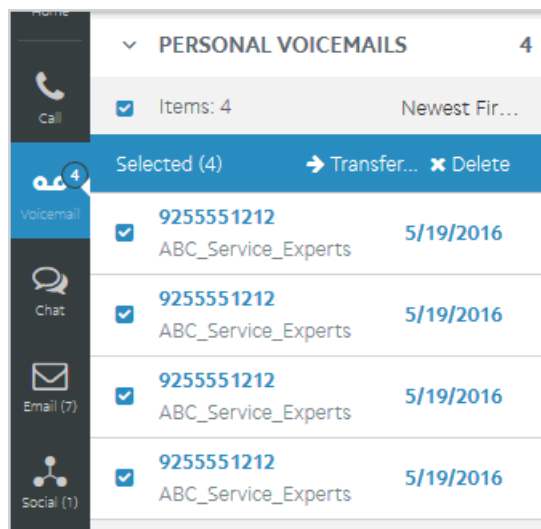
- **Keep as Recording** – Associates the recording with the call and makes it available for reporting.
- **Send to Agent** – Routes the recording to the agent that receives the call, as voicemail (not Hangup Module).
- **Delete** – Deletes the recording.

There are two purposes for using audio files:

- **Call post-processing** – When the information cannot be gathered using DTMF or speech recognition (name, street address, etc.), it can be recorded using the Voice Input module and delivered to the agents. The recordings are transformed into voicemail with an ability to be delivered as skill voicemail.
- **Regular recordings** – The callers input is saved as audio files. Agents will not have an access to the files in this case. This option can be used if you do not need to convert the recorded information into the written form. For example, regular recordings can be useful for surveys. The recordings can also help

administrators to debug their IVR scripts and fine-tune the choices configured in a Menu module.

If the **Send to Agent** option is selected in the IVR script, the recording is available to the agent who can access the recordings in the **Voicemail** tab. Recordings are labeled with the incoming skill queue.

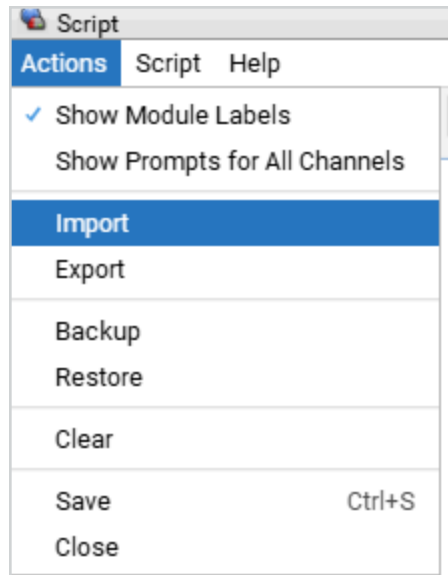


Assuming the IVR Script is set to **Keep as Recording** mode, recorded audio files gathered via IVR modules are transformed into recordings (not voicemail). They are available via Reports and saved on FTP server with a unique file name based on the campaign's applicable recording file naming convention, appended with **_<Recording ID>**. The **Voice Input** module name is a reference of the audio file that holds the recording.

Importing Scripts

IVR scripts are located in the navigation pane. You can import a script in an existing script, or in a new script. Double-click the script.

- 1 In the IVR workspace, select **Actions** > **Import**.



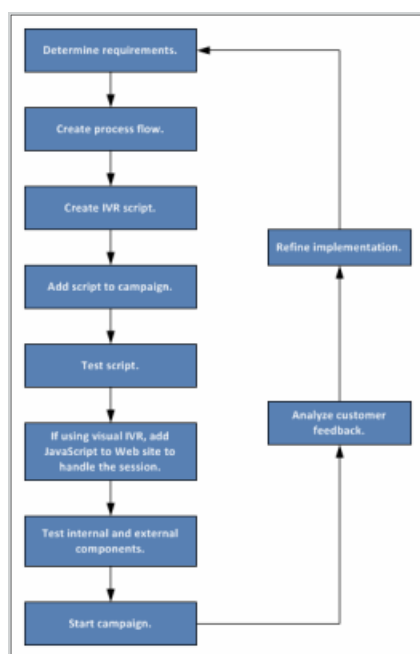
- 2 To see the configuration, double-click each module in the imported script.

Designing IVR Scripts

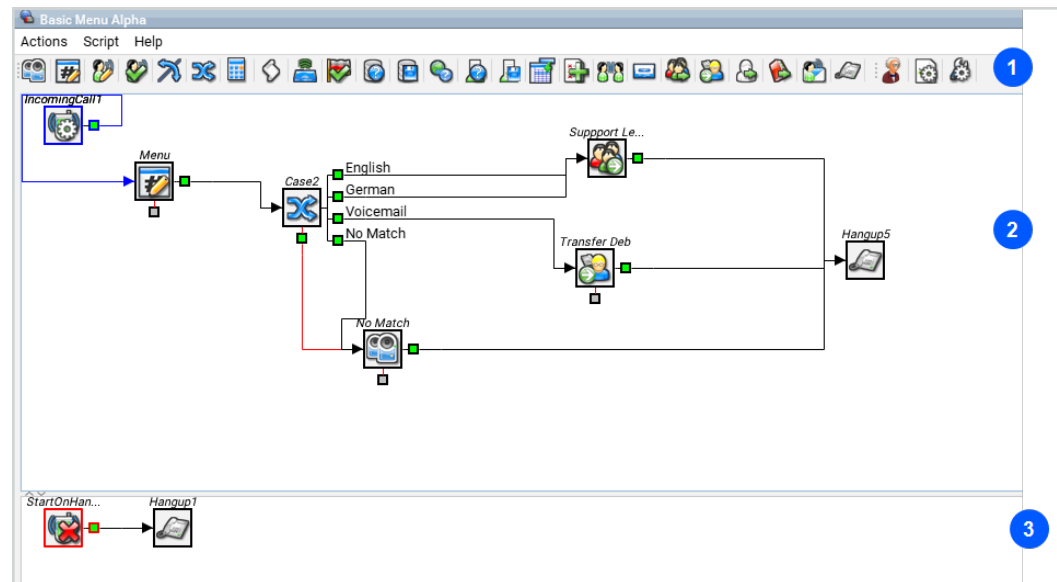
You create IVR scripts to contain diverse components, such as speech recognition, linked modules, and variables.

[IVR Script Designer](#)
[Speech Recognition Grammars](#)
[IVR Modules](#)

IVR Script Designer



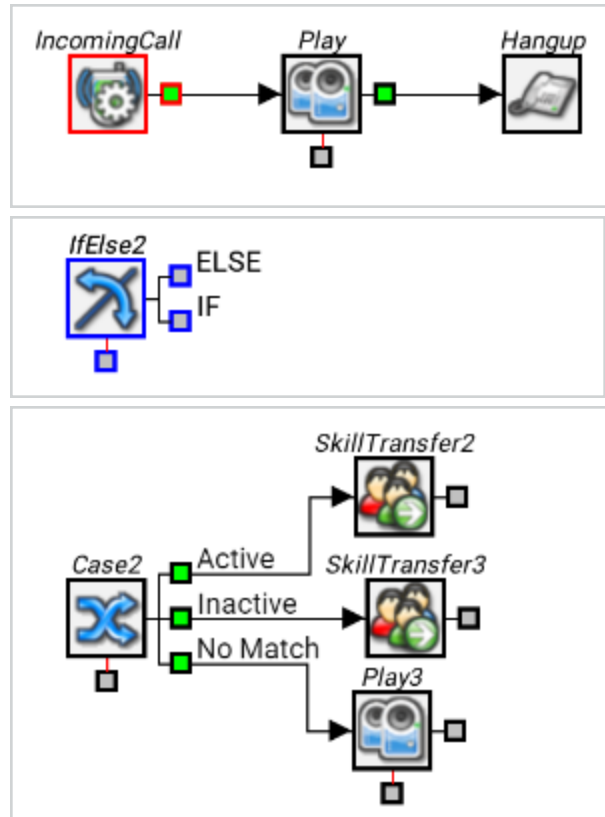
To access the designer, select **IVR Scripts > Edit**.



- 1 Modules toolbar –IVR module icons that you drag to the work area. Hover over the icon to view the module name.
- 2 Work area – Re-sizable window where you construct your script with modules and connectors.
- 3 Bottom pane – Post-disconnect handling. For example, with the Query Module and Web2Campaign, abandoned callers can be added to an outbound campaign for a callback. The fields of a contact record can be updated with Abandoned=TRUE, which would identify records to include in an outbound campaign (functions in reporting can automatically add these contacts to a list).

Each module contains an icon, label (can be hidden), and connector ports. Module relationships and order are presented as arrows which connect the icons. Each module has properties.

IVR modules are connected to each other by connector ports as shown in the figure below. Several modules can follow one module (IncomingCall module). Some modules have two output ports (If/Else and Answering Machine modules). Some modules have one output for each branch (Case module). Termination modules have no output ports.



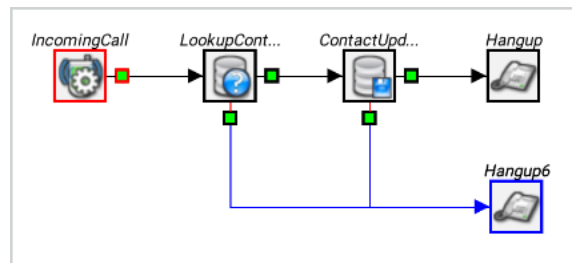
The module execution can cause errors and exceptional situations. For example, a request for external data may not be answered for a specified timeout, a contact record may not be found in the database, etc. For handling these exceptional conditions, there is an additional error exit from the modules. Most of the modules have an exception port.

For example, the Voicemail Transfer module can have these exceptions: Transfer to voicemail cannot be executed and Wrong agent or skill information in Variable (Agent/Skill can not be found). The Hang Up module does not have any exceptions.

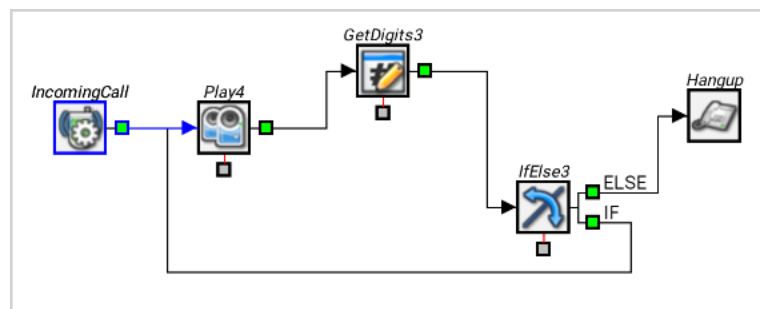
Using the exception port is optional. If no action is specified for the exception, it is ignored and the main exit is used.

ERR_CODE variable contains the value from the previous module. The error codes can differ for each IVR module. Error codes can be found in topics describing a particular IVR module.

You can link one branch into another branch in order not to repeat the same modules for each branch.



You can form loops by connecting a module to a previous module. A loop with exit by condition (like WHILE) looks like:

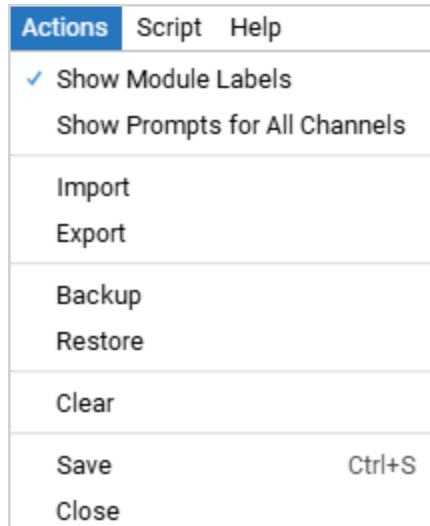


A loop, such as FOR, with a predefined number of iterations requires that you define a loop counter variable and use an expression in a Set Variable module to increase or decrease it. See also [Creating an IVR Loop](#).



Actions Menu

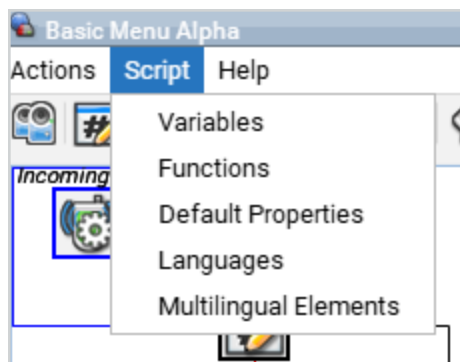
In the script, select **Actions** to see the available menu choices.



- Importing, exporting, restoring, and backing up: Your IVR Scripts can be exported to create backups of currently-running scripts or to save scripts that are not yet completely configured. IVR Scripts and all supporting subscripts and prompts can be backed up and restored. See [Exporting and Importing IVR Scripts](#).
- Clear: Enables you to remove all elements from the working area except the Incoming Call icon.
- Save: Validates your script. If no errors exist, the script is saved and you can close the designer. If errors exist, you see an error message.

Script Menu

In the script, select **Script**.



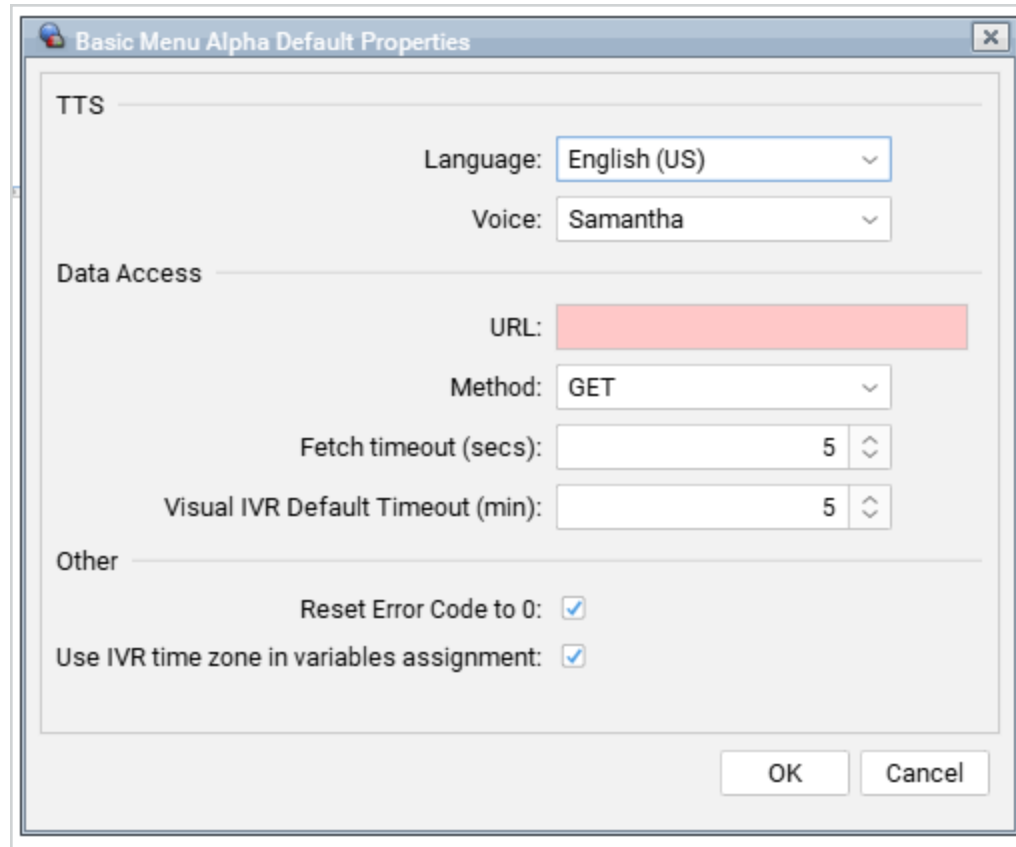
[Default Properties](#)
[Variables](#)
[Languages](#)
[Multilingual Elements](#)

Default Properties

The IVR Script properties window can be accessed by selecting the View Properties of option from the IVR script right-click menu in the IVR Scripts folder. The General tab contains this information:

- IVR Script Name: Not editable.
- Description: Text area that identifies the purpose of the script. The description is shown on the IVR Scripts screen together with the script name.
- Max time in IVR: Setting to prevent unintended infinite loops configured in an IVR script from keeping calls in the IVR for an indefinite amount of time.
- Associated Campaigns: View all campaigns that use this script. You can't remove a script if any campaigns are using it.
- Called By Scripts: View all IVR scripts that run this script.

You can configure the default properties of the Text-To-Speech editor and for data access.



The image shows a dialog box titled "Basic Menu Alpha Default Properties". It contains three sections: "TTS", "Data Access", and "Other".

- TTS:**
 - Language: English (US) (dropdown)
 - Voice: Samantha (dropdown)
- Data Access:**
 - URL: (text field, highlighted in red)
 - Method: GET (dropdown)
 - Fetch timeout (secs): 5 (spin box)
 - Visual IVR Default Timeout (min): 5 (spin box)
- Other:**
 - Reset Error Code to 0: ☒
 - Use IVR time zone in variables assignment: ☒

At the bottom right are "OK" and "Cancel" buttons.

- TTS:
 - Language: List of languages that you can choose for TTS prompts in the Text-To-Speech editor: English (UK), English (US), Spanish (MX), and Portuguese (BR).
 - Voice: Voice character available to speak the voice prompts. The voice is female.
- Data Access: Default parameters used to access the data in the Query module.
 - URL: Address of the data.
 - Method: GET and POST methods.
 - Fetch timeout: Time-out in seconds for the request.
 - Script Type: Type of script for Web server: ASP, PERL, PHP.
 - Script path: Location of the script on your system.
 - Visual IVR Default Timeout (min): Time-out in minutes for the visual IVR to appear.

Important: Five9 recommends setting this to the 30-minute maximum for IVR scripts used to transfer chat interactions to chatbot.

- **Reset Error Code to 0:** Resets `__error_code__` value to 0 between interconnected modules.
 - **Enabled:** The `__error_code__` value is set in each executed module. Default value for new scripts.
 - **Disabled:** The `__error_code__` value carried throughout execution.

Example

Consider three subsequent IVR modules: Module1 and Module2 lead to an IfElse module. The ifElse module starts with an `__error_code__` value of 0. If an exception happens in Module1, the `__error_code__` of Module1 is set to a corresponding non-zero value. Module2 is completed normally. In this example, the IfElse goes to the TRUE branch as soon as the Module2 is executed without problems. Otherwise, the IfElse module returns FALSE.

- **Use IVR time zone in variables assignment:** Select this option to use the IVR [Set Variable Module](#) to assign IVR `_TIME_` variable to `Contact.Time`. Conversely, you can use this option with the [Contact Update Module](#) to synchronize the contact variables with the contact record.

Variables

A variable is a field to use in your IVR script to store and manipulate temporary data in the IVR. Variables have a name, type, and value. Variables are essential parts of a script. All script modules can access defined variables. Store call-attached data in the form of variables to use in latter parts of the IVR script or integrate into the text-to-speech messages. You can assign customer data to variables.

Use the predefined variables or create your own to store temporary data. You manage variables in the Variables window opened in the IVR Script properties window (the Script menu). See also, [Managing Variables](#).

While creating TTS prompts, enter text phrases and insert contact, call, or system variables in the text. See also, [Adding Prompts](#).

Use variables to hold values to keep for later processing during script execution. When you request the external Web server for information, you will probably want to use the result of that request somewhere else in your script.

Variables include these types:

- **User Variables:** Variables have local scope and are accessible only by the script where they were defined. The BUFFER variable is automatically created in each script. Create script variables with names that include alphanumeric characters and underscores, starting from a letter or underscore.
- **Call Variables:** Metadata about a call: Call.ANI, Call.DNIS, Call.bill_time, Call.campaign_id, Call.campaign_name, Call.comments, Call.disposition_id, Call.disposition_name, Call.end_timestamp, Call.handle_time, Call.hold_time, Call.length, Call.number, Call.park_time, Call.queue_time, Call.session_id, Call.skill_id, Call.skill_name, Call.start_timestamp, Call.type, Call.type_name, and Call.wrapup_time.
- **Call Attached Variables (CAVs):** Key-value pairs specific to calls available until the agent assigns a disposition. These variables are accessible from the IVR by the agent who receives the call or transfer, by agents who are consulted or added to a conference call, and from the IVR again when a call transfers back to the IVR. Call variables are global variables that are often permanent (stored for reporting purposes); whereas IVR variables are temporary local variables used within an IVR script.
- **System Variables:**
 - **CRM_ID:** Internal ID of the CRM record currently selected in the IVR.
 - **DATE:** System date set in the time zone of the campaign that contains the IVR.
 - **DAY:** Numeric value that represents the day of the week set in the time zone of the campaign that contains the IVR.
 - **1: Sunday**
 - **2: Monday**
 - **3: Tuesday**
 - **4: Wednesday**
 - **5: Thursday**

- **6: Friday**
- **7: Saturday**
- **INPUT:** Optional variable to use as follows:
 - **DTMF:** Retains the caller's choice for further branching.
 - **Speech recognition:** Words spoken by the caller. Use this variable to add a confirmation prompt, for example, *You have chosen {apples}. Is this correct?*
- **LAST_AGENT_EXTENSION:** Phone extension of the last agent who handled the call.
- **LAST_AGENT_NAME:** User name of the last agent who handled the call.
- **SWI_LITERAL:** Words spoken by the caller as interpreted by the last Input module.
- **TIME:** System time set in the time zone of the campaign that contains the IVR.
- **TIME_ZONE:** Time zone of the campaign that contains the IVR.
- Contact Variables (same as Contact Data Fields, see [Basic Configuration Administrator's Guide](#), [Contact Data Fields](#))
- Agent Variables (Agent.first_agent, Agent.full_name, Agent.id, Agent.user_name)
- IVR Variables (IVR.error_code, IVR.error_desc, IVR.last_module, IVR.last_state)

Data Types. Each IVR variable has a type. Use types to simplify data consistency control.

String	Boolean*
Number	Percent
Date	Email
Date/Time	URL
Currency	Phone Duration
* The only boolean values accepted are lowercase <code>true</code> and <code>false</code> .	

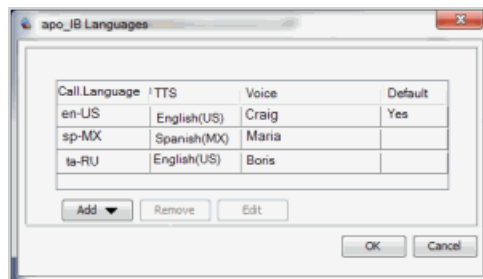
Values. The following are ways of assigning values to a variable while using IVR:

- By specifying initial value in variables definition screen. See [Managing Variables](#).
- From scheduler (for upper level script). See [Managing IVR Script Schedules for Autodial and Inbound Campaigns](#) in the [Campaign Administrator's Guide](#).
- From calling procedure (if the script is invoked by other script). See [Foreign Script Module](#).
- By using the Set Variable IVR module.
- As user input.
- Through internal database dip (for contact variables).
- In result of request to Web server.

Variables of any type not initialized by any of these options have a special NULL value. It is possible to assign NULL value to a variable or check if a variable is NULL using the If/Else module. NULL values aren't defined for logical expressions and results in exceptions in corresponding modules (If/Else and Case modules).

Languages

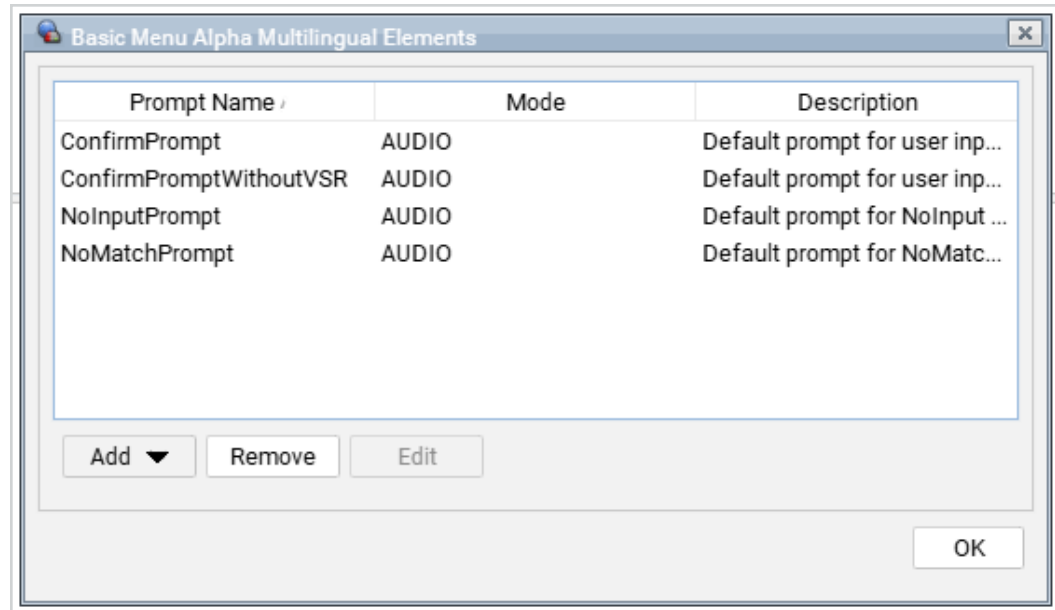
In the IVR script builder, select **Actions > Script > Languages** to specify a set of supported languages for specific IVR scripts.



You must have a Call Language variable for each language in your IVR script, as shown above. Select **Add** to specify the default **TTS** language and **Voice** for new prompts for a new Call Language variable. You can set any the TTS to any language, but you must set a default language.

Multilingual Elements

You can create dynamic audio for your scripts: visuals, text, menu items, and estimated wait time announcements. To add items, select **Script > Multilingual Elements**.



Audio prompts. Click **Add** and create or use an existing prompt. The menus are similar to other prompt menus.

Visual prompts. Click **Add** and create or use an existing prompt. The menus are similar to other prompt menus.

Text. Click **Add** to add text and variables.

Menu item. Click **Add** to create audio, visual, and text-based prompts and other items containing variables and thumbnail images.

EWT announcement. Plays an estimated wait time (EWT) message to the customer stating when an agent will answer a call transferred by the Skill Transfer module. The EWT announcement plays after the set standard prompt (see [Adding Prompts](#)).

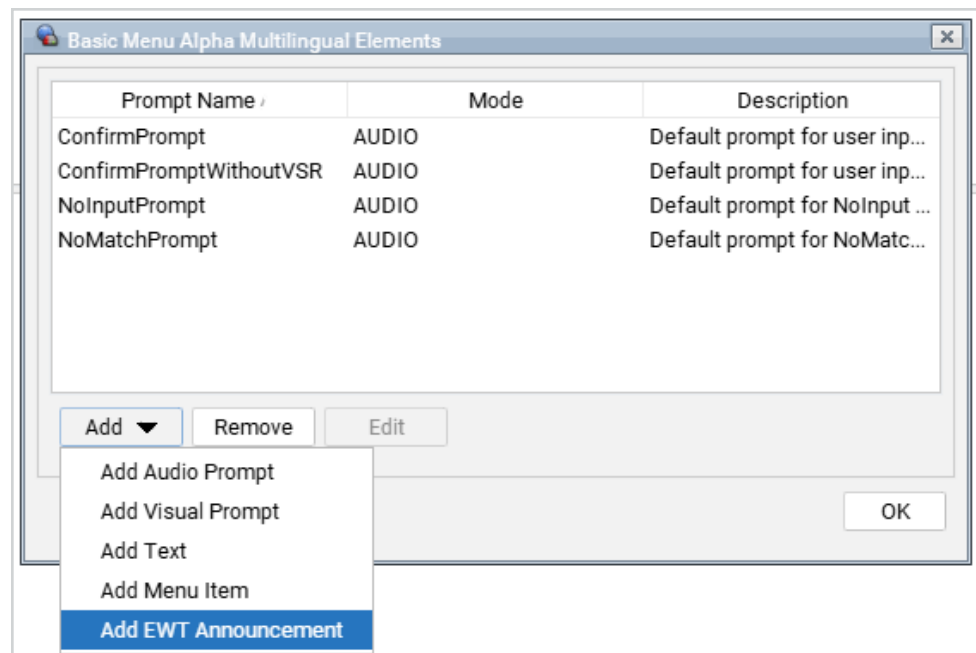
When creating an EWT announcement, the EWT interval (in seconds) is evaluated as:

- EWT interval > 0 : If between 0–59 seconds, < 1 minute
- EWT interval > 1 : If between 60–119 seconds, > 1 minute
- EWT interval > 2 : If between 120–179 seconds, > 2 minutes

The EWT *interval* is the number of accepted calls transferred by the Skill Transfer module in the last 15 minutes divided by the total queue time.

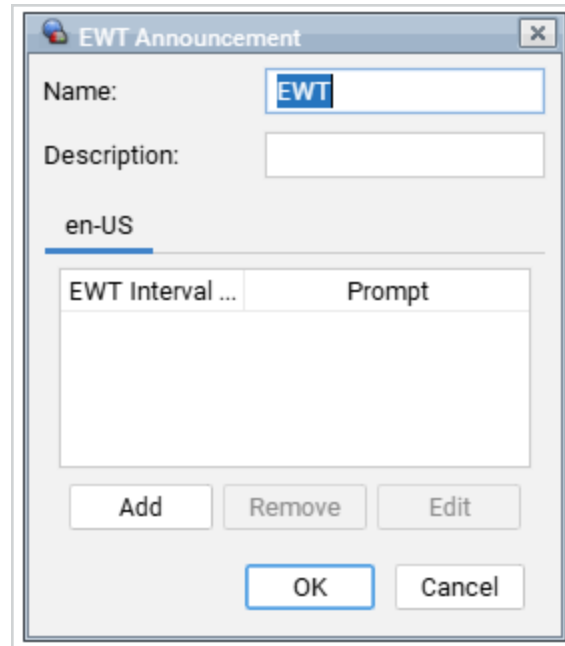
Note: If the call is already assigned to an agent, EWT is always 60 seconds.

1 Click **Add, and select **Add EWT Announcement**.**



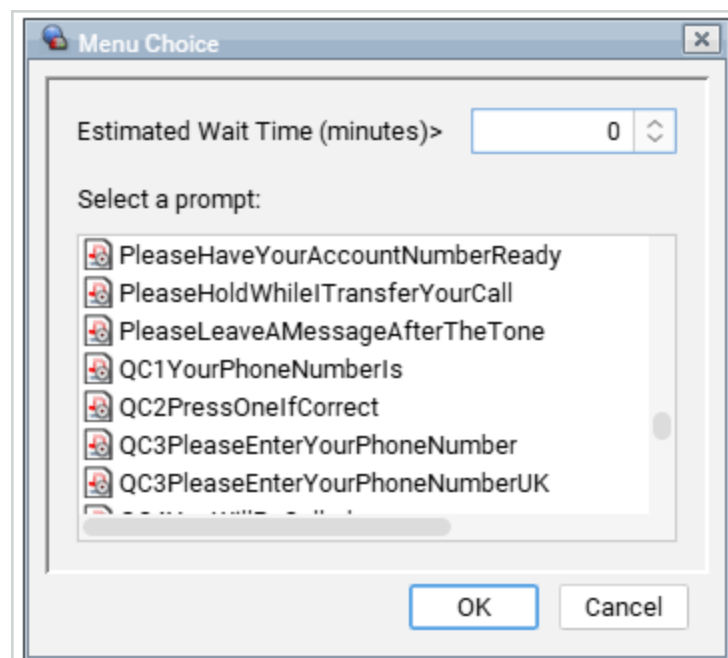
2 Name and describe the announcement.

The name displays on the tab in the EWT Announcement window.



3 Select **Add**.

The Prompt menu contains a list of announcements you created for different EWT durations or events (see [Adding Prompts](#)).



- 4 Select a prompt and the EWT to send in the announcement.
- 5 Select **OK** two times.

Speech Recognition Grammars

Note: If you need speech recognition features, contact your Five9 account director.

These grammar types only apply if you have the Five9 IVR with Speech Recognition package to create advanced self-service voice applications, for example:

- Menu-driven call routing, like with a customer prompt: *You can say Sales or Customer Service.*
- Entering an account or tracking number.

The standards for speech recognition grammars are located at <https://www.w3.org/TR/speech-grammar/>.

The **Input** and **Menu** modules use speech recognition to compare spoken words to a predefined grammar of expected phrases from a variety of voices:

alphanum	digits
boolean	number
ccexpdate	phone
creditcard	socialsecurity
currency	time
date	zipcode

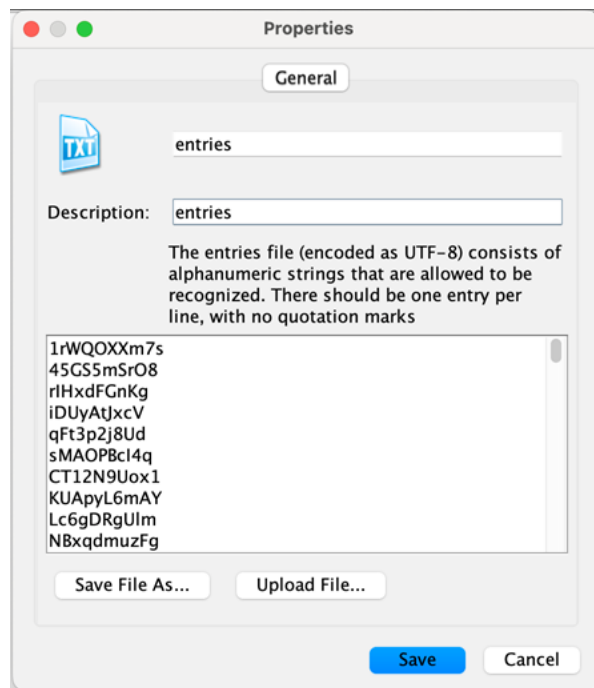
alphanum

Recognizes alphanumeric strings (a connected string of up to 20 digits and alphabet characters), such as *abc123*. Use this grammar to recognize items such as, a product code or order number. alphanum isn't case sensitive. Allowed characters are A–Z and a–z, and the digits are 0–9. Non-alphanumeric characters, such as hyphens (-), dots (.),

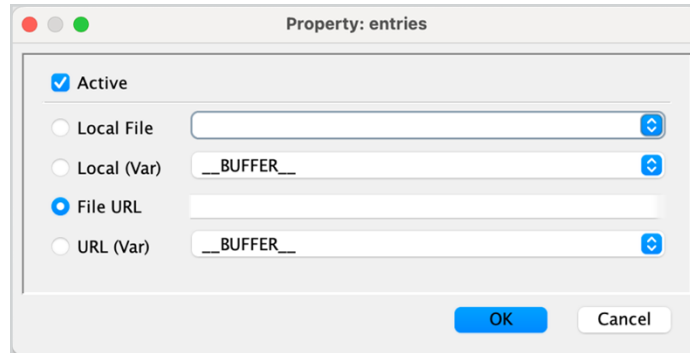
and underscores (_) aren't recognized; when spoken, these reduce recognition accuracy.

Parameter properties.

Parameter	Description
entries*	URI specifying a file listing allowed constraints, one per line.
* <i>Required</i>	



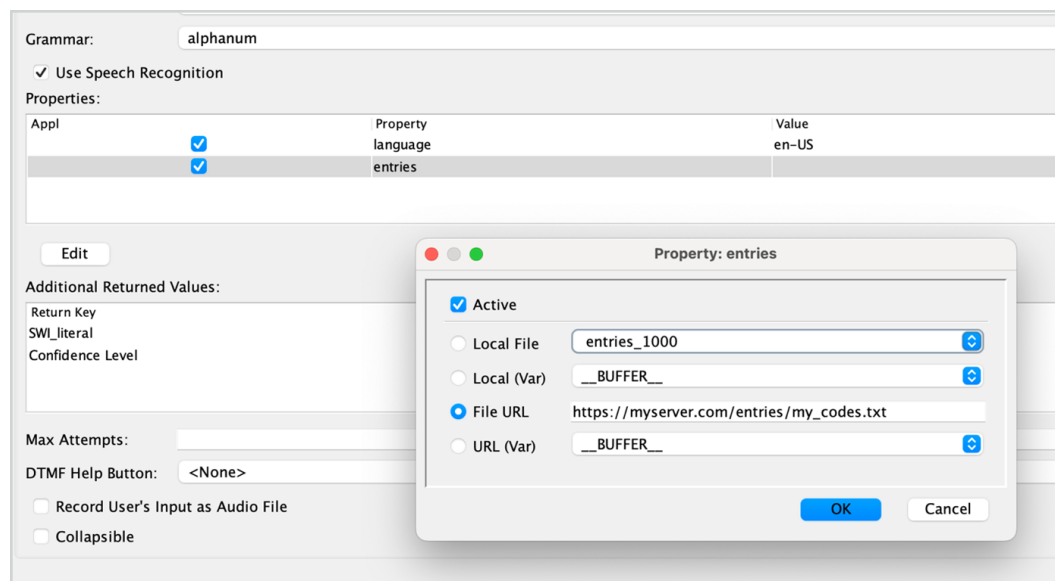
The possible options for `entries` are **Local File** or **File URL**. The **Local File** option allows you to use files uploaded to **Misc IVR Files** in the IVR Scripts node. You can use variables for both options.



Files can contain over a million entries, but must be smaller than 1M. An error displays if you try to upload a file larger than 1M.



File URL allows you to use a remote file that can have over a million entries and be retrieved using the URL.



The `entries` file (encoded as UTF-8) consists of alphanumeric strings to recognize, one entry per line. Don't use quotation marks. Constraints are uppercase and lowercase alphabetic characters (recognition considers cases the same). Non-alphanumeric characters, such as dashes or spaces, are ignored. The entries can't contain special characters: those with accents (for example, è) or tildes (~). By default, each constraint has no spaces between the elements.

You can add a header with a list of directives to the top of the file. The header must consist of lines beginning with a double colon (::) followed by a directive (no spaces).

Directive	Description
::DELIMITED	This file uses spaces between each element in each constraint. Spaces are ignored. Use this directive when two characters represent a single alphabet character (for example, the Spanish letter <i>ch</i>).
::WEIGHTED <i>max</i>	Constraints have a weight value following the entry to indicate the probability of hearing that constraint (for example, 12a4 and 1284 sound similar in English). You can assign a higher weight to the entry that would more likely be spoken. <i>max</i> is the maximum weight a constraint can have. When using <code>WEIGHTED</code>, you must end every constraint with <code><space><real_number></code> where, <i>real_number</i> is the weight to assign the constraint. Note: - You must include the space even if not specifying ::DELIMITED. - Non-numeric weights cause the entries file to fail to load. Any specified value greater than <i>max</i> defaults to <i>max</i> and a warning displays.
::LENGTH <i>len</i>	All constraints in the file are the same length, <i>len</i> characters long. This directive improves recognition accuracy.
::NUM <i>n</i>	This file contains <i>n</i> constraints (or fewer). This can speed the loading of the entries file as only the specified number of constraints will load.

Note: Dual tone multi-frequency (DTMF) is not recognized by `alphanum`.

Return key/values.

Key	Description
MEANING	Contains a string of ISO-8859-1 digits and lowercase letters with no embedded spaces.

Additional Returned Key/Values.

Key	Description
SWI_literal	Contains the exact recognized text.

This is an example entries file:

You can limit the list of constraints to numbers. For example, a *digits.txt* entries file might contain:

1564

2345

8642

1119999911

boolean

Accepts yes or no from the caller. *Correct* is accepted as a synonym for *yes*. By default, 1 (=yes) or 2 (=no) is assigned to the variable.

Parameter properties.

Parameter	Default Value	Description
y	1	Desired DTMF digit equivalent to yes.
n	2	Desired DTMF digit equivalent to no.

Return key/values.

Key	Description
MEANING	True or false.

Additional Returned Key/Values.

Key	Description
SWI_literal	Exact text that was recognized.

ccexpdate

Understands the expiration date on a credit card. Expiration dates are usually a month and a year embossed on a credit card in the form mm/yy or MonthYYYY. The grammar recognizes variations of the date, for example, December 2007, twelve oh seven, twelve of two thousand and seven, twelve slash zero seven, etc.

Parameter properties.

Parameter	Default Value	Description
referencedate	current date	Date in YYYYMMDD format to use in computing ranges.
maxallowed	84 months	Latest date recognized expressed as a number of months beyond reference date.
maxexpected	48 months	Latest likely date recognized expressed as a number of months beyond reference date.

Return key/values.

Key	Description
MEANING	Recognized date in YYYYMMDD format. For example, 20110331 refers to March 31, 2011. The value is the same whether or not the caller

Key	Description
	specified a day of the month. The day is always set to the last day of the month. Can be assigned to a selected variable in the Input Module (default is BUFFER).

creditcard

Understands a credit card number, optionally preceded by the credit card name or the words *account number* or *account*. For example, a caller can say, *Visa account number four oh one seven...*, *MasterCard five zero zero two...*, or *three seven three five....* The grammar understands any commercial credit card that adheres to the ISO standard numbering scheme: all cards issued by Visa, MasterCard, American Express, Discover, and Diners Club, and many private label cards provided by individual retailers (department stores, gas companies, etc.).

This standard dictates minimum and maximum length of the digit string, open digit sequences, and a check digit used as the last digit in the string so that the entire number meets the mod 10 checksum criterion. Five9 recommends that you validate numbers with related information (Expiration date, Security codes, Database validation).

Parameter properties.

Parameter	Default Value	Description
<code>typesallowed</code>	All types are recognized.	Credit card types. To specify more than one, use the list to enable individual card types: American Express, Diners Club, Discover, MasterCard, Private Label, and Visa.

Return key/values.

Key	Description
MEANING	Recognized credit card number. Can be assigned to a selected variable in the Input Module (default is BUFFER).

Additional Returned Key/Values.

Key	Description
CARDTYPE	Assigned to the type recognized, where the types are of the same set that can be specified in the type property. Even if the caller does not specify the card type, the type is inferred from the card number, for example: all Visa cards start with 4.

currency

The currency grammar is available for these currencies:

- Dollars and cents: For speech recognition, these phrases are accepted: *ten dollars*, *ten dollars and fifteen cents*, and *ten fifteen*.
- UK pound and pence
- Euro and cents

You may use all available currencies in all the relevant parts of your scripts: grammars, display, prompts, and input modules, such as Get Digits, Input, and Set Variable.

All currency, IVR, system, and call variables support the UK and Euro currencies.

Parameter properties.

These parameters apply only to speech recognition for US English only. They do not apply to TTS.

Parameter	Default Value	Description
minallowed	0.0	Minimum amount.
maxallowed	9999.99	Maximum amount.
minexpected	0.0	Minimum likely amount.
maxexpected	9999.99	Maximum likely amount.
granularityallowed	0.01	Granularity of recognized results.
granularityexpected	0.01	Granularity of likely results.

Return key/values.

Key	Description
MEANING	String in this format: <currency><main unit amount><dot><subunit amount>. Automatically assigned to the SWI_literal variable in the Input module.

Additional Returned Key/Values.

Key	Description
SWI_literal	Exact text that was recognized.

date

Accepts a date spoken in any of several formats, such as mm-dd-yyyy or mm/dd/yyyy.

Parameter properties.

Parameter	Default Value	Description
minallowed	19000101	Earliest date recognized, expressed as YYYYMMDD
maxallowed	21991231	Latest date recognized
minexpected	19000101	Earliest likely date
maxexpected	21991231	Latest likely date

Return key/values.

Key	Description
MEANING	Date string expressed as YYYYMMDD. Recognized phrases include 4 June, 4 June 2006, 4, 6, 2006, the 4th, 4th June, and Monday, the 4th of June. The MEANING key is automatically assigned to the SWI_literal variable in the Input module.

Additional Returned Key/Values.

Key	Description
SWI_literal	Exact text that was recognized.

digits

Recognizes a continuously spoken string of up to 20 digits (that is, the caller is not required to pause after each digit). Valid characters are the digits 0-9. The digit 0 can be pronounced as either *oh* or *zero*. Punctuation characters such as hyphens (-), dots (.), and underscores (_) are not recognized; if spoken they reduce recognition accuracy.

Parameter properties.

Parameter	Default Value	Description
minlength	1	Minimum number digits recognized
maxlength	20	Maximum number digits recognized

Return key/values.

Key	Description
MEANING	String of digits (ISO-8859-1 encoding), with no spaces, for example: 12345. Automatically assigned to the SWI_literal variable in the Input Module.

Additional Returned Key/Values.

Key	Description
SWI_literal	Exact text that was recognized.

number

Accepts quantities such as *ten*, *one hundred and forty*, *five hundred sixty one point five*, *negative five*, and *minus four point three*.

Parameter properties.

Parameter	Default Value	Description
<code>minallowed</code>	0	Minimum number.
<code>maxallowed</code>	999 999 999.99	Maximum number.
<code>maxdecimal</code>	2	Maximum number of decimal places (maximum 9).
<code>minexpected</code>	0	Minimum likely number.
<code>maxexpected</code>	999 999.99	Maximum likely number.
<code>granularityallowed</code>	1.0	Granularity of recognized results. This must be an integer. If set, fractional entries are recognized. To control allowed decimal precision, use the <code>maxdecimal</code> parameter.
<code>granularityexpected</code>	1.0	Granularity of likely results.

Return key/values.

Key	Description
MEANING	String of ISO-8859-1 digits with no spaces, for example: if the caller says <i>twenty five</i> , the result is 25. Automatically assigned to the <code>SWI_literal</code> variable in the Input Module.

Additional Returned Key/Values.

Key	Description
<code>SWI_literal</code>	Exact text that was recognized.

phone

Accepts 7- and 10-digit North American phone numbers and three-digit numbers ending in 11, such as 911. An optional 1 may be placed before 7- or 10-digit numbers. Dashes are optional. For example, if 16789999 is recognized, the returned result is 6789999. The grammar allows phrases, such as *three two four fifty five seventy two* and strings of individual digits.

Return key/values.

Key	Description
MEANING	Variable-length string of digits with no spaces. Automatically assigned to the SWI_literal variable in the Input Module. Assigned to a string of digits representing the recognized phone number. A leading 1 is omitted in the return value. For example, if 16789999 is recognized, the return result is 6789999. The return string may optionally contain the character x to indicate a phone number with an extension, for example: 8005551234x789.

Additional Returned Key/Values.

Key	Description
SWI_literal	Exact text that was recognized.

socialsecurity

Understands 9-digit US Social Security numbers. For example, a caller can say, *oh one seven, four seven, six five eight three*. Illegal numbers, such as those beginning with three zeros, are rejected. The grammar accepts numbers with first digits ranging from 0 to 7 but does not accept those beginning with 8 or 9.

Return key/values.

Key	Description
MEANING	Recognized number. Can be assigned to a selected variable in the Input Module (default is BUFFER).

time

Time of day spoken by the caller, for example: HH:MM [AM | PM]. Recognized phrases include times given in 12-hour format (5 o'clock), 24-hour format (twenty-three fifteen), and qualified times such as *before 5 o'clock* and *about 5*.

Parameter properties.

Parameter	Default Value	Description
minallowed	0000	Earliest time recognized.
maxallowed	2359	Latest time recognized.
minexpected	0000	Earliest likely time.
maxexpected	2359	Latest likely time.
granularityallowed	1	Allowed granularity in minutes.
granularityexpected	1	Expected granularity in minutes.

Return key/values.

Key	Description
MEANING	Four-digit string in the 24-hour format, from 0000 to 2359. Automatically assigned to the SWI_literal variable in the Input Module.

Additional Returned Key/Values.

Key	Description
SWI_literal	Exact text that was recognized.

zipcode

Recognizes valid United States ZIP Codes in either five- or nine-digit format.

Return key/values.

Key	Description
MEANING	Recognized five- or nine-digit ZIP code. Automatically assigned to the SWI_literal variable in the Input module.

Additional Returned Key/Values.

Key	Description
SWI_literal	Exact text that was recognized.

IVR Modules

Each IVR module controls an aspect of a call. An IVR script is a set of linked modules. When the system receives a call, the IVR script routes the caller from one module to the next until the caller is transferred to an agent, voicemail, or another destination. Specific modules are used at the start, middle, or end of a script. For a list of error messages that you can see with each module, see [Error Messages](#).

Start Module**Center Modules**

[Get Digits Module](#)[Play Module](#)[Voice Input Module](#)

Transfer and Terminating Modules

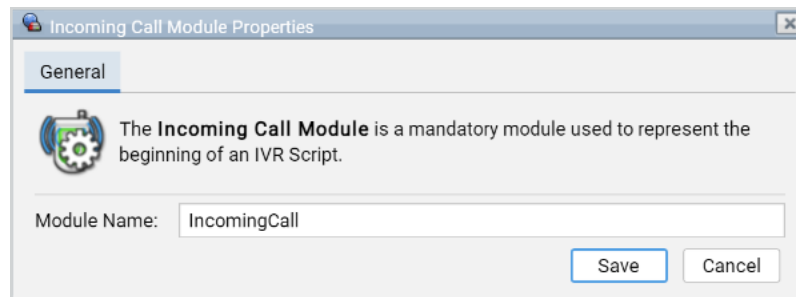
[Extension Transfer Module](#)[Voicemail Transfer Module](#)[Start on Hangup Module](#)[Third Party Transfer Module](#)[Hangup Module](#)

Important

Incorrect regular expressions in IVR modules are no longer executed for a long time. They are now stopped after one second.

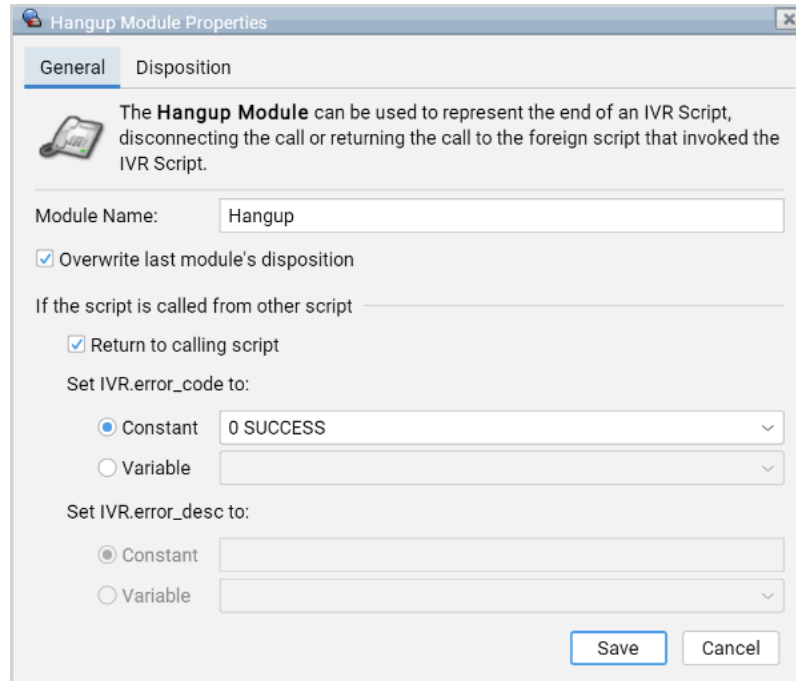
Incoming Call Module

The Incoming Call module is the first module of all IVR scripts. By default, it is already in the work area when you create a IVR script. You cannot remove this module from the work area. You can name the module, but you cannot assign properties to it. See [IVR Modules](#) for the description.



Hangup Module

Use this module to disconnect a call, which ends the script. If the script was started from another IVR script, this module returns the call to the initial script.



The screenshot shows the 'Hangup Module Properties' dialog box with the 'General' tab selected. The dialog has a title bar with a close button. Below the title bar are two tabs: 'General' and 'Disposition'. The 'General' tab contains a description of the Hangup Module, a text field for 'Module Name' (set to 'Hangup'), a checked checkbox for 'Overwrite last module's disposition', a section for 'If the script is called from other script' with a checked checkbox for 'Return to calling script', and two sections for setting error codes: 'Set IVR.error_code to:' with 'Constant' selected and '0 SUCCESS' in the dropdown, and 'Set IVR.error_desc to:' with 'Constant' selected and an empty text field. At the bottom right are 'Save' and 'Cancel' buttons.

Hangup Module Properties

General Disposition

The **Hangup Module** can be used to represent the end of an IVR Script, disconnecting the call or returning the call to the foreign script that invoked the IVR Script.

Module Name:

☒ Overwrite last module's disposition

If the script is called from other script

☒ Return to calling script

Set IVR.error_code to:

☒ Constant

☐ Variable

Set IVR.error_desc to:

☒ Constant

☐ Variable

Save Cancel

General tab.

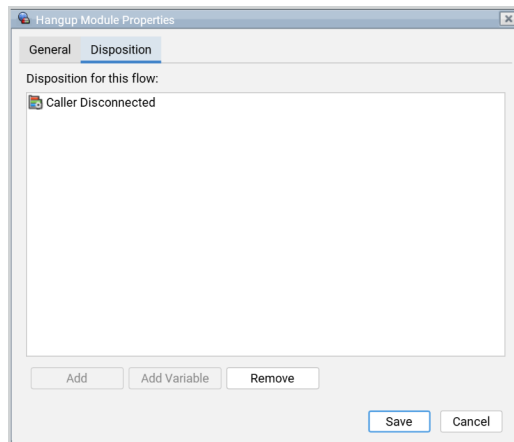
- **Module Name:** Name of the module specific to the script or the campaign.
- **Overwrite last module's disposition:** If a disposition was assigned in the previous module in the script, this option enables you to change the disposition for the call.
 - Enables the Disposition tab.
- **If the script is called from other script:** If the current script follows another script, select how to exit the current script:
 - Return to calling script: continue with the parent IVR script.
 - Set IVR.error_code to:
 - Constant: Specify the error code to return to the parent script.
 - Variable: Specify the contact, system, IVR, or call variable to use for the error code that is returned to the parent script.

Disposition tab.

- **Caller Disconnected:** Default disposition.
- **Add:** Select this button to add a disposition.

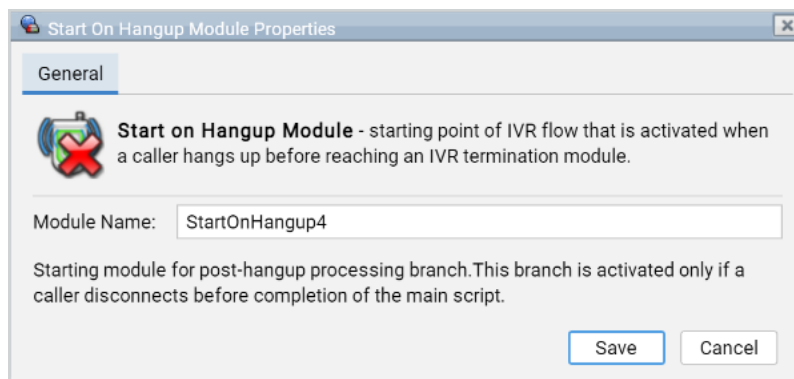
- **Add Variable:** Select this button to add a variable that contains the value of the disposition you want to set.
- **Remove:** Select this button to remove a disposition.

For more information about dispositions, see [Selecting Dispositions](#).



Start on Hangup Module

This module enables you to start another process when callers hang up before reaching a termination module, a transfer module, or the Hangup module. You do not need to configure this module.



Troubleshooting an IVR Script

Note: The VCC Debugger does not support loopback numbers. These numbers are used for internal calls within the same domain and are not reachable outside of Five9.

Use the Five9 VCC Debugger to troubleshoot an IVR script. To access the debug tool, do the following:

- 1** Navigate to the **Campaign properties DNIS** tab.
- 2** Click **Set Params > Start**.
The Five9 VCC Debugger screen opens.
- 3** Select the session type and system variable, and enter the value.
- 4** Click **Start in Browser** to launch.
When the debug tool is launched, it will be triggered when the information entered is matched. Once the information is matched, the debugger will navigate the IVR script and verify whether or not the script is accurate.
- 5** If an error is found, the module(s) with an error is highlighted in red. Click the highlighted module name to display the details in the browser.
The debugger will run through the modules included in the script. Some modules contain more information than others.

Modules

The following are the modules outlined in [IVR Modules](#). For a list of error messages that you can see with each module, see [Error Messages](#).

Agent/Voice Mail Transfer Module	Menu Module
Answering Machine Module	Natural Language Processing Module
Case Module	Play Module
Conference Module	Query Module
Contact Update Module	Salesforce Module
Extension Transfer Module	Set DNC Module
Five9 Rules Engine Module	Set Variable Module
Foreign Script Module	SFDC Einstein Module
Get Digits Module	Skill Transfer Module
If/Else Module	System Info Module
Input Module	Third Party Transfer Module
Iterator Module	Update System Info Module
IVA Transfer Module	Voice Input Module
Language Module	Voicemail Transfer Module
Lookup Contact Record Module	

Agent/Voice Mail Transfer Module

The Agent/Voice Mail Transfer module is a termination module that enables you to transfer calls to agents or to voicemail.

If **Agent Transfer** is selected then the interaction is routed to the specified agent.

Note

You can also route by agent extension id or by variable. Additionally, Five9 supports the module's **transfer to any available agent** option.

Agent Assignment for Text Channel Interactions

The following rules apply to agent assignment for text channel interactions.

Transferring to a Specific Agent

An interaction is assigned to an agent when the agent identified by the agent transfer module is:

- Logged in
- Ready
- Available
- Not at interaction capacity

Transferring to Any Available Agent

To transfer to any available agent, select to enable the **Transfer to any available agent** option. The text interaction is routed to any available agent, chosen using longest-wait algorithm. If there are no available agents then the interaction is routed to the default skill for the profile. Because you may not use call variables that contain sensitive information in this field, these variables do not appear in the menu even if they exist in the Call Variables folder of the VCC.

For text channel interactions, Five9 stops processing the IVR script after using an agent transfer module regardless of agent availability.

If a user abandons the chat session while waiting for an agent to accept that interaction, it is not assigned to the agent selected by the agent transfer module.

The Text Channels Administrator Console setting **Ignore Agent Interaction Limits for Transfers and Conferences** does not apply to interaction limits for this module. The text interaction is assigned to an agent only if the current interaction capacity is less than the maximum allowable capacity, regardless of this option setting in SCC.

Agent/Voicemail Assignment for Voice Interactions

This module enables voice interaction transfer to agent, skill, or voicemail options to exit an IVR script.

[General Tab](#)
[Agent Tab](#)
[Voice Mail Tab](#)
[Prompts Tab](#)
[Dispositions Tab](#)

General Tab

The screenshot shows the 'Agent/Voice Mail Transfer Module Properties' dialog box with the 'General' tab selected. The dialog has a title bar and a close button. Below the title bar are tabs for 'General', 'Agent', 'Voice Mail', 'Prompts', and 'Disposition'. A descriptive text block states: 'The Agent/Voicemail Transfer Module can be used to queue callers and play in-queue hold music and announcements until the callers are routed to a specific agent or agent's voicemail, or the Default skill.' Below this is a 'Module Name' field containing 'AgentTransfer3'. A section titled 'Select one of the following functionalities:' contains two radio buttons: 'Transfer to any available agent' (selected) and 'Agent Transfer'. Below this are three time input fields: 'Max Queue Time (mm:ss)' set to '01:00', 'Max Ring Time (secs)' set to '15', and 'Pause Before Transfer (secs)' set to '0'. There are three checked checkboxes: 'Queue Calls When Agent(s) Not Ready', 'Set agent to Not Ready if no answer', and 'Enable Music On Hold'. At the bottom is an 'Action for Recorded Files' dropdown menu set to 'Keep as Recording'. 'Save' and 'Cancel' buttons are at the bottom right.

Agent/Voice Mail Transfer Module Properties

General Agent Voice Mail Prompts Disposition

The Agent/Voicemail Transfer Module can be used to queue callers and play in-queue hold music and announcements until the callers are routed to a specific agent or agent's voicemail, or the Default skill.

Module Name: AgentTransfer3

Select one of the following functionalities:

☒ Transfer to any available agent

☐ Agent Transfer

Max Queue Time (mm:ss): 01:00

Max Ring Time (secs): 15

Pause Before Transfer (secs): 0

☒ Queue Calls When Agent(s) Not Ready

☒ Set agent to Not Ready if no answer

☒ Enable Music On Hold

Action for Recorded Files: Keep as Recording

Save Cancel

- **Transfer to any available agent:** Enables the Voice Mail tab.
- **Agent transfer:** Enables the Agent tab.
- **Max Queue Time:** Maximum number of minutes and seconds allowed for calls in a queue.
- **Max Ring Time:** Maximum number of seconds allowed for agents to answer calls.
- **Pause Before Transfer:** Number of seconds before transferring a call to a selected agent.
- **Queue Calls When Agent(s) Not Ready:** Whether to add calls to a queue for the maximum queue time specified when agents are not available:
 - Enabled: Calls are placed in a queue.
 - Disabled: Calls are transferred immediately to voicemail.
- **Set Agent to Not Ready If No Answer:** Whether to change an agent's status to Not Ready if the agent does not answer the call within the Max Ring Time.
 - Enabled: Agent's status is changed.
 - Disabled: Agent's status is not changed.
- **Enable Music On Hold:** Whether to play an audio file when the caller is waiting in the queue.

- **Action for Recorded Files:** Method for handling the recorded files gathered by the previous modules. See also [Using Recorded Files](#).
 - Keep as Recording: Save recorded files as a recording to access from reports. The recorded files are not sent to the agents or skills.
 - Send to Agent: Routes recorded files to the agent. If agents are not available and the voicemail option is enabled, the recorded files become part of the voicemail message.

Agent Tab

Enabled if you selected **Agent Transfer** in the General tab. Transfers the call to either and agent or to an extension.

The screenshot shows the 'Agent/Voice Mail Transfer Module Properties' dialog box with the 'Agent' tab selected. The 'Agent Name' section has two radio buttons: 'Constant' (selected) and 'Variable'. Below them are two text input fields. The 'Extension' section also has two radio buttons: 'Constant' (selected) and 'Variable'. Below them are two text input fields, with the first containing '0000'. At the bottom, there is a checkbox labeled 'Transfer to any agent if specified one is unavailable' which is currently unchecked. 'Save' and 'Cancel' buttons are at the bottom right.

- **Agent Name:** Agent who receives the transfer after the maximum queue time expires:
 - Constant: List of your agents.
 - Variable: List of your contact, system, agent, call, IVR, or user CAV variables. Only string variables are allowed.

Note

If the assigned agent is unavailable, such as logged out, not ready, or at capacity, and **Transfer to any agent** option is not enabled, the system will seek alternatives in this order:

- 1 Default agent specified in system variable.
- 2 Default skill specified in system variable.

- **Extension:** Extension who receives the transfer after the maximum queue time expires:
 - Constant: Specific four-digit extension.
 - Variable: List of your contact, system, agent, call, IVR, or user CAV variables.
- **Transfer to any agent if specified one is unavailable:** Whether to transfer the call to any available agent if necessary.
 - Enabled: Transfer the call to any available agent.
 - Disabled: Seek transfer to defaults specified above (Agent Name).

Voice Mail Tab

Enabled if you selected **Transfer to any available agent** in the General tab.

The screenshot shows the 'Agent/Voice Mail Transfer Module Properties' dialog box with the 'Voice Mail' tab selected. The 'Leave voicemail' checkbox is checked. Under the 'Agent Name' section, the 'Constant' radio button is selected, and the text field contains a dropdown arrow. Under the 'Extension' section, the 'Constant' radio button is selected, and the text field contains '0000'. There are 'Save' and 'Cancel' buttons at the bottom right.

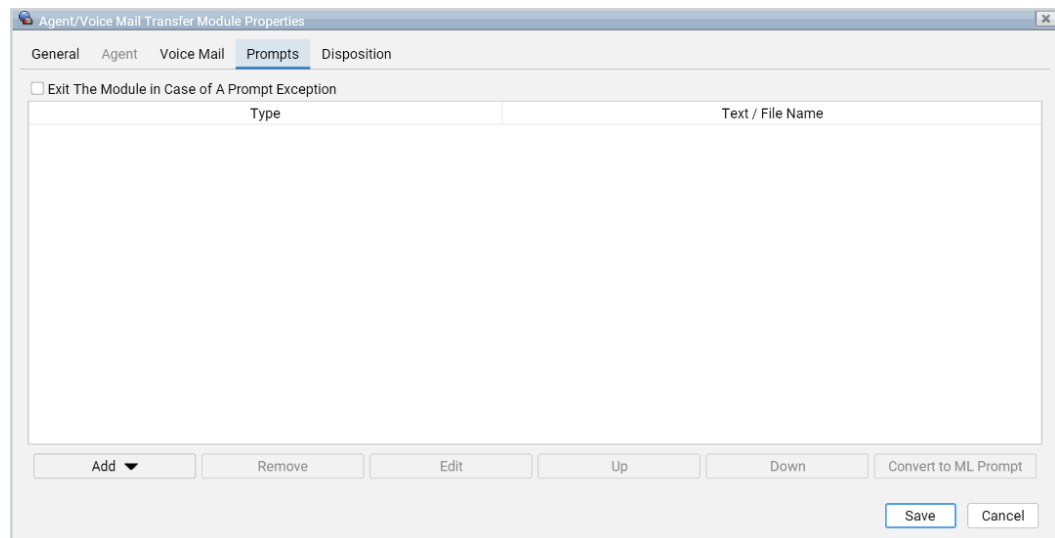
Typically used to send calls to a general voicemail user account. You can create a user account to capture general messages after selecting a skill group in the Skill Transfer module.

- **Leave voice mail:** Whether to enable voicemail messages for the agent or extension.
- **Agent Name:** Voice mailbox to receive the transfer after the maximum queue time expires:
 - Constant: List of all your agents.
 - Variable: List of your contact, system, agent, call, IVR, or user CAV variables. Only string variables are allowed.

- **Extension:** Extension to receive the transfer after the maximum queue time expires:
 - Constant: Specific four-digit extension.
 - Variable: List of your contact, system, agent, call, IVR, or user CAV variables.

Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds).



- **Exit the module in case of a prompt exception:** Whether to exit the module if the prompt cannot be played.
 - Enabled: Exits the module when the prompt cannot be played.
 - Disabled: Does not exit the module. The defective prompt is ignored.

Dispositions Tab

The default disposition is Abandon. For more information, see [Selecting Dispositions](#).

Answering Machine Module

The Answering Machine module is used to determine if the call is answered by a person or a machine and to route calls according to the script.

Important: If you plan on requiring this functionality, the only supported location is the first module after the Incoming Call Module. Using the Answering Machine Module later in the flow is not supported and may trigger error 1400.

The parts of the module are used as follows:

- **Answering machine:** Module branch used when the system determines that the call is answered by the recording of an answering machine. In this case, the script proceeds to the next module, which is usually a Play module that contains a recorded message followed by a Hang Up module.
- **Live person:** Module branch used when the system determines that the call is answered by a person. In this case, the script proceeds with the rest of the script.

If you omit this module, the beginning of the recorded message of the following [Play module](#) is cut off while the answering machine greeting is played.

The screenshot shows the 'Answering Machine Module Properties' dialog box with the 'General' tab selected. It contains a description of the module's function, a text field for the 'Module Name' set to 'AnswerMachine2', a checked checkbox for 'Wait for Answering Machine Tone', and a 'Maximum Wait Time (secs)' spinner set to 20. 'Save' and 'Cancel' buttons are at the bottom right.

Answering Machine Module Properties

General

The **Answering Machine Module** can be used to automatically determine whether an answering machine has answered a call, and indicate to the IVR Script whether the call was answered by a machine or a live person.

Module Name: AnswerMachine2

☒ Wait for Answering Machine Tone

Maximum Wait Time (secs): 20

Save Cancel

- **Wait For Answering Machine Tone:** Whether to wait for a tone. This option is typically used when the answering machine tone may not be immediately detected.
 - Enabled: Wait for a tone
 - Disabled: Do not wait for a tone.

Important: When this option is enabled, the system waits the configured maximum wait time, anticipating the tone at the end of the voicemail greeting. After the configured maximum wait time, the system proceeds to the next module, typically a Play module with a recorded message. In a case where the answering machine message exceeds the configured maximum wait time, the recorded message may be

abbreviated or not heard at all, or the call may be disconnected while the voicemail greeting is still playing.

- **Maximum Wait Time:** Applies if you enabled the answering machine tone option. Number of seconds to wait for the answering machine greeting. The default is 20 seconds.

Case Module

The Case module enables you to create as many branches as you need to handle all the possible user input or the information contained in a variable. Each possible case must have its own branch.

Example

A Get Digits module contains this prompt: *For Sales, press 1. For Support, press 2.* The Get Digits module is followed by the Case module that contains the BUFFER variable with the values 1, which is leads to the Sales branch, and 2, which is leads to the Support branch. In this case, the Case module has three branches: Sales, Support, and No Match to handle all other caller input.

For more example, see also [Creating an IVR Script](#) and [Creating an IVR Loop](#).

Case Module Properties

General

The **Case Module** can be used to select one of many multiple branches in the IVR Script logic based on the value of a variable or input previously provided by a caller.

Module Name:

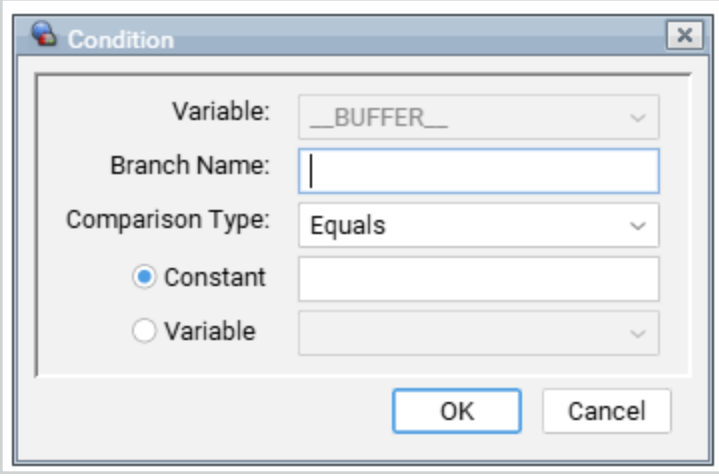
Variable:

Comparison Type	Value/Variable	Branch Name
Equals	"1"	Sales
Equals	"2"	Support
		No Match

- **Variable:** Variable that stores the caller's input, for example when the Case module follows the Get Digits module. Use the default variable (BUFFER), select one from your contact, system, agent, call, IVR, or user CAV variables, or create custom variables in the **Script > Variables** menu. You can add, modify, delete, and order variables as needed. Represents the left variable in the equation.
- **Table:** Comparison Type Value/Variable Branch name: Represent the operator, the right variable, and the exit from the module, for example:

Variable	Comparison Type	Value/Variable	Branch name
BUFFER	equals	1	Sales

- **No Match branch:** Default branch used when the comparison does not match the available options. For example, if the prompt specifies to press 1 or 2, but the caller presses 3 or nothing, the script uses the No Match branch to exit the module.
- **Condition window:** Menu that enables you to define the right variable of the equation.



- **(Left) Variable:** Always BUFFER.
- **Branch Name:** Exit from the module. Branch names can contain up to 50 characters.
- **Comparison Type:**
 - Equals: Exact match
 - Like: Supports wildcards % and _.
 - RegExpr: Regular expression that matches specific characters.
 - IsNULL: Empty field

- **(Right) Variable:**
 - Constant: Specific value that you want to find.
 - Variable: List of your contact, system, agent, call, IVR, or user CAV variables.

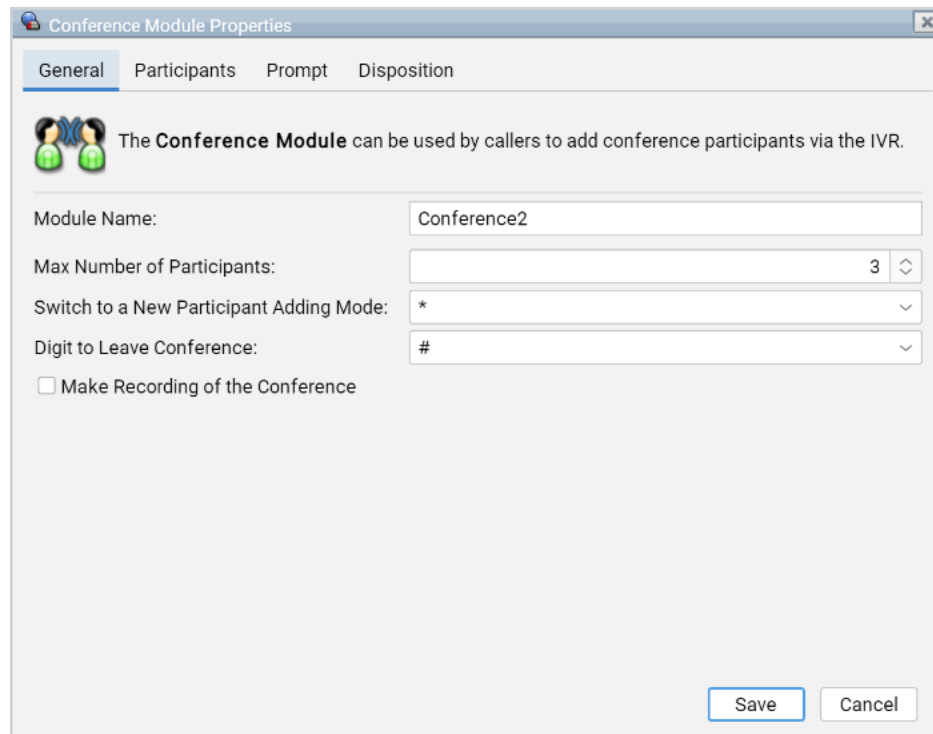
Conference Module

The Conference module enables callers to add other parties to a conference call. Callers entering this module are prompted to enter a phone number to invite a participant. The module dials the number and connects the second party to the first caller. Participants can leave the conference at any time. Except while adding a conference participant, the first caller can leave the conference or hang up. At that time, the conference is discontinued, and all participants are disconnected.

If your domain is not enabled for the E.164 format for international phone numbers, agents must enter 10 digits to conform to the [North American Numbering Plan](#). If your domain is enabled for the E.164 format, agents must enter phone numbers for all calls according to your configuration: in your national format or up to 15 digits, starting with the plus (+) sign and the country code. If you do not specify a format, the VCC uses the E.164 format for all numbers. For more information, see the [Basic Configuration Administrator's Guide](#).

[General Tab](#)
[Participants Tab](#)
[Prompt Tab](#)
[Dispositions Tab](#)

General Tab



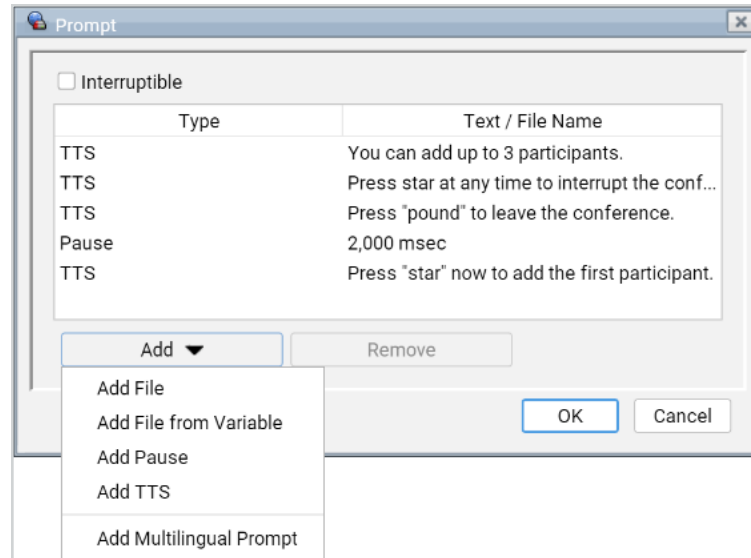
The screenshot shows the 'Conference Module Properties' dialog box with the 'General' tab selected. The dialog has four tabs: 'General', 'Participants', 'Prompt', and 'Disposition'. Below the tabs is a header area with an icon of three people and the text: 'The **Conference Module** can be used by callers to add conference participants via the IVR.' Below this are four configuration fields: 'Module Name' (text box with 'Conference2'), 'Max Number of Participants' (spin box with '3'), 'Switch to a New Participant Adding Mode' (dropdown menu with '*'), and 'Digit to Leave Conference' (dropdown menu with '#'). At the bottom left is a checkbox labeled 'Make Recording of the Conference' which is currently unchecked. At the bottom right are 'Save' and 'Cancel' buttons.

- **Max Number of Participants:** Maximum allowed number of conference participants. If the number is exceeded, add a prompt on the Prompt tab to inform the person who is managing the conference.
- **Switch to a New Participant Adding Mode:** DTMF button to access the conference option to enter a new participant phone number for the conference. The default is the star (*) key.
- **Digit to Leave Conference:** DTMF button to exit the conference. The default is the pound (#) key.
- **Make Recording of the Conference:** Whether to save the conference as audio file. The recording is available in reports. See also [Using Recorded Files](#).

Participants Tab

- **Prompt:** Default prompt comprising two TTS prompts. You can add prompts from a file, a variable, or the TTS builder. You can arrange the prompts in any

sequence. Between prompts, you can add brief pauses (milliseconds).



- **Interruptible:** Whether to stop playing the prompt as soon as the requested operation is completed.
 - Enabled: Stop playing the prompt.
 - Disabled: Finish playing the prompt in all cases.
- **Max Number of Attempts:** Number of attempts to enter the phone number of a new participant.
- **Events:** Error messages and actions requested from users to resolve the problem. You can add prompts from a file, a variable, or the TTS builder. You can arrange the prompts in a specific sequence.

The screenshot shows the 'Conference Module Properties' dialog box with the 'Prompt' tab selected. The dialog has four tabs: General, Participants, Prompt, and Disposition. The 'Prompt' tab is active, showing settings for 'Entering a New Participant's Phone Number'. It includes a 'Prompt' text box with the text 'Please enter a phone number. When connected, press *star* to add the conference part...', a 'Max Number of Attempts' spinner set to 3, and an 'Events' table. The table has three columns: Event, Event's Prompt, and Action. It lists two events: 'NoMatch' with prompt 'NoMatchPrompt' and action 'reprompt', and 'NoInput' with prompt 'NoInputPrompt' and action 'reprompt'. There is an 'Edit' button below the table. At the bottom, there are two dropdown menus: 'Return to Conference with New Participant' set to '*' and 'Reject to Add a New Participant' set to '#'. 'Save' and 'Cancel' buttons are at the bottom right.

Event	Event's Prompt	Action
NoMatch	NoMatchPrompt	reprompt
NoInput	NoInputPrompt	reprompt

Prompt Tab

Prompts that are played during different stages of the conference. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). All prompts can be interruptible.

Conference Module Properties

General Participants **Prompt** Disposition

On Module Entering

Prompt:

You can add up to 3 participants. Press star at any time to interrupt the conference for a... Edit

If No Participants Left

Prompt:

There are no participants in the conference at the moment. Press "star" to add a particip... Edit

If Number of Participant exceeded

Prompt:

The allowed number of participants is exceeded. Edit

Save Cancel

- **On Module Entering:** Default TTS prompt played when a user enters the conference. This prompt comprises four TTS prompts and a pause.
- **If No Participants Left:** Default TTS prompt played for the person who created the conference when all other participants have left the conference. The prompt is repeated every 10 seconds if the caller does not enter a key. After three attempts, the module is exited with an exception.
- **If Number of Participants Exceeded:** Default TTS prompt when the maximum number of participants has been reached. The menu item for interruptible prompts is disabled.

Dispositions Tab

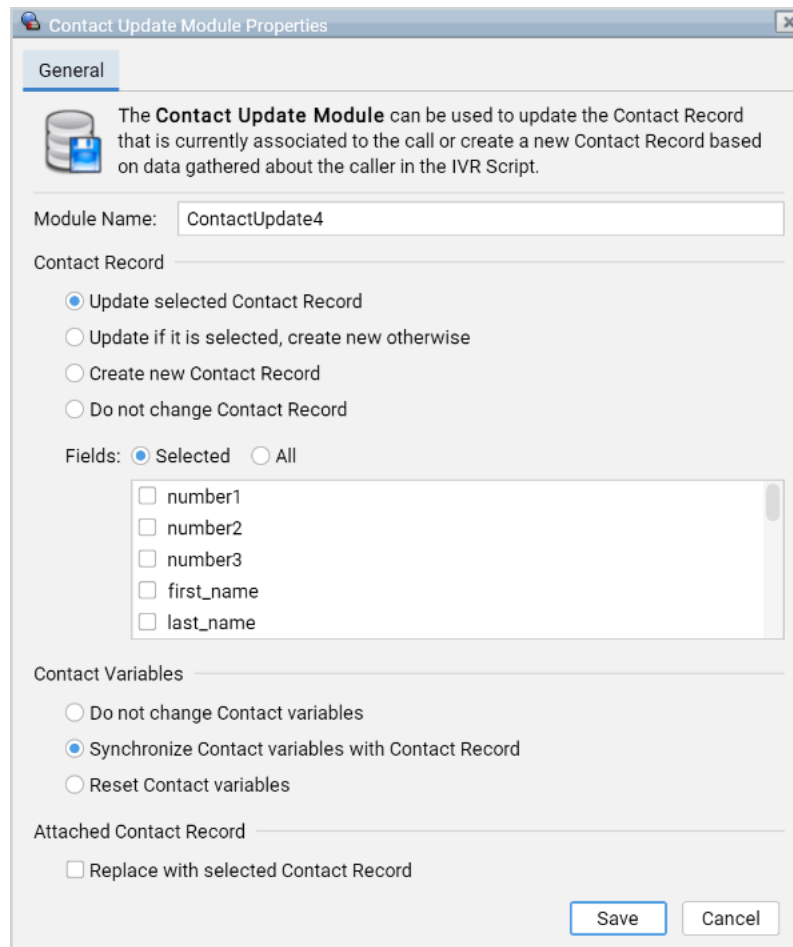
The default disposition is No Disposition. For more information, see [Selecting Dispositions](#).

For example, you can add dispositions to handle these events:

- A prompt is repeated if the caller does not press a key for 10 seconds. After three attempts, the module is exited with an exception.
- The caller is disconnected while the module is being processed.

Contact Update Module

The Contact Update module enables you to use the current call data to create or update a contact record and to attach an existing contact record to the call.



The screenshot shows the 'Contact Update Module Properties' dialog box with the 'General' tab selected. The dialog contains the following sections:

- General:** A description states: 'The **Contact Update Module** can be used to update the Contact Record that is currently associated to the call or create a new Contact Record based on data gathered about the caller in the IVR Script.'
- Module Name:** A text field containing 'ContactUpdate4'.
- Contact Record:** Four radio button options: 'Update selected Contact Record' (selected), 'Update if it is selected, create new otherwise', 'Create new Contact Record', and 'Do not change Contact Record'.
- Fields:** A radio button group with 'Selected' (selected) and 'All'. Below is a list box containing checkboxes for 'number1', 'number2', 'number3', 'first_name', and 'last_name'.
- Contact Variables:** Three radio button options: 'Do not change Contact variables', 'Synchronize Contact variables with Contact Record' (selected), and 'Reset Contact variables'.
- Attached Contact Record:** A checkbox labeled 'Replace with selected Contact Record' which is currently unchecked.
- Buttons:** 'Save' and 'Cancel' buttons at the bottom right.

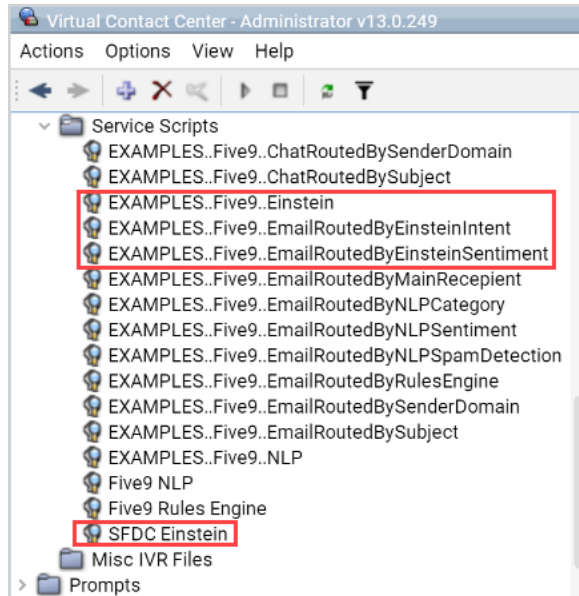
- **Module Name:** Name that represents the role of the module in the IVR script. In Visual IVR scripts, the name is converted to text that appears in the heading of the IVR window.
- **Contact Record:** Action to take with the current call data of the record:
 - **Update Selected Contact Record:** Update the record with the call data by using the contact fields below.

- Update if it is selected, create new otherwise: update by using the contact fields below.
 - Create New Contact Record: Create a contact record by using the contact fields below. A new record is considered a current record and CRM_ID is changed accordingly.
 - Do not change Contact Record: You cannot change the record because the contact fields below are disabled.
- **Fields:** List of contact fields that you can use to update the contact record in the first three options above.
- **Contact Variables:** Action to take with the contact variables:
 - **Do not Change Contact Variables:** Makes no update to the contact variables.
 - **Synchronize Contact variables with Contact Record:** Enables you to set values from the variables if the corresponding fields are not checked for update.
 - **Reset Contact Variables:** Enables you to assign default values, such as an empty string, or 0, for all variables.
- **Replace with Selected Contact record:** Whether to replace the attachment with another contact record:
 - Enabled: Replaces the attachment with a contact record that you define.
 - Disabled: Maintains the original contact record.

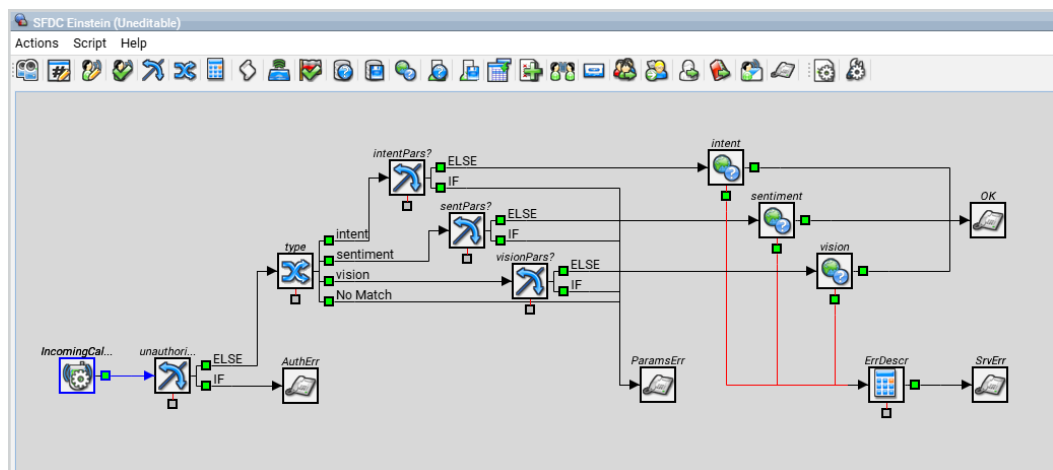
SFDC Einstein Module

The SFDC Einstein module enables you to integrate your implementation with the Salesforce Einstein Platform Service technologies, including Einstein Vision and Einstein Language. This section describes how to use Einstein Language to identify sentiment and intent to provide enhanced routing decisions and recommended next best actions to your agents. To use SFDC Einstein, you need to train the Einstein engine to your requirements. For more information about Einstein training model development, features, and capabilities, see the [Salesforce Einstein documentation](#).

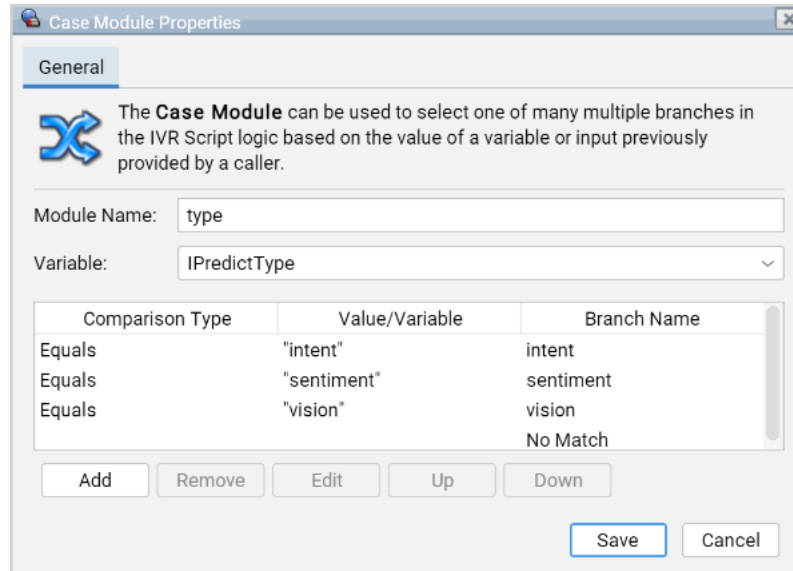
You can access the SFDC Einstein script for non-editable viewing at **IVR Scripts > Service Scripts > SFDC Einstein**. To edit the files, right-click the script name and select make a copy, then rename and save or go directly into the IVR builder.



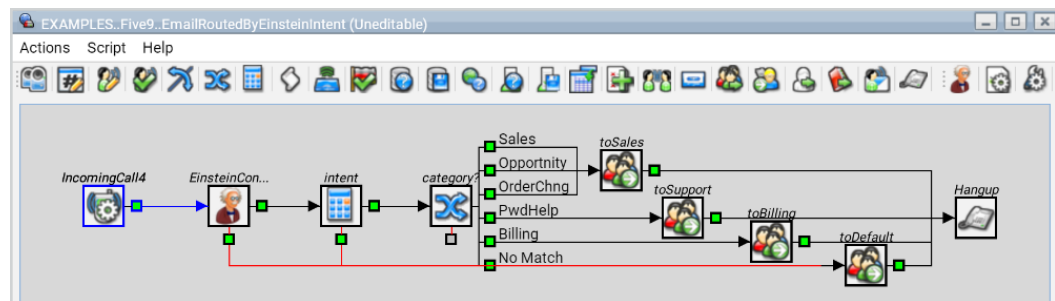
The SFDC Einstein module is a pre-packaged IVR script within a module; the example file is shown below.



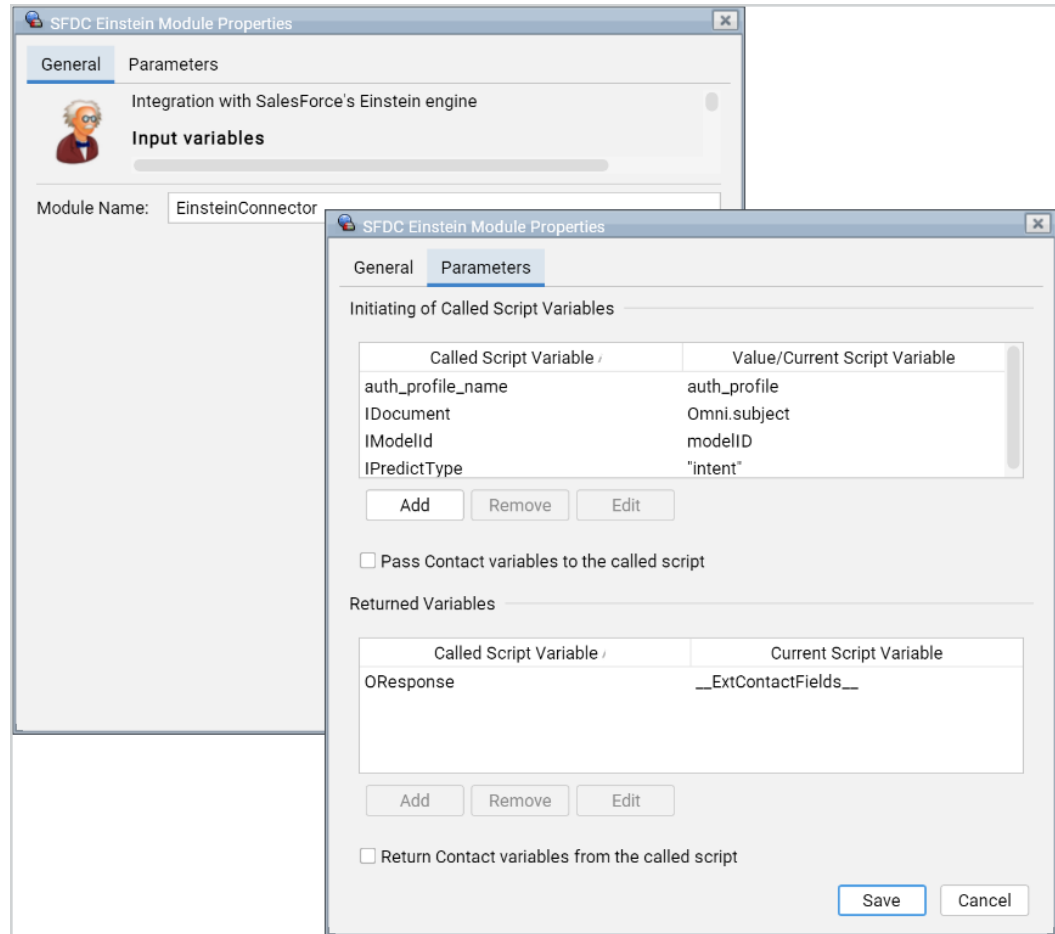
You can view each module used within the script. For example, opening the properties for the first Case module within the SFDC Einstein script shows the distribution for sentiment, intent, and vision.



You can drag the SFDC Einstein module directly to your canvas, or create a duplicate of a prepared script and substitute your domain-specific information. The example below shows a script designed to route by SFDC Einstein-determined intent.



Once you have made a duplicate script, or otherwise introduced the SFDC Einstein module to your IVR script, right-click or double-click to access the module properties.

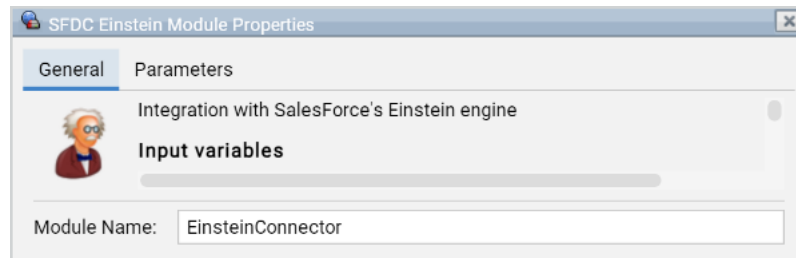


You can rename the module in the **General** tab and assign your specific parameters in the **Parameters** tab.

Properties

[General Tab](#)
[Parameters Tab](#)

General Tab

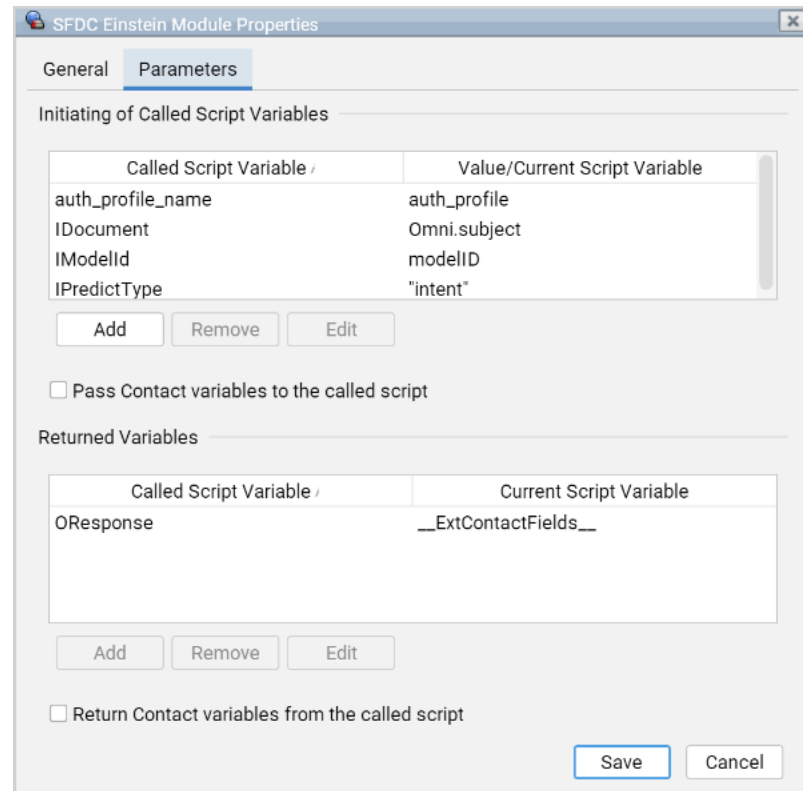


The screenshot shows the 'General' tab of the 'SFDC Einstein Module Properties' dialog. It features a header with 'General' and 'Parameters' tabs. Below the header is a section titled 'Integration with Salesforce's Einstein engine' with a toggle switch. Underneath is a section titled 'Input variables' with a progress bar. At the bottom, there is a 'Module Name' field containing the text 'EinsteinConnector'.

- **Module Name:** Name shown in IVR builder.

Parameters Tab

- **Input Parameters:** Contains called script with variable parameters:



The screenshot shows the 'Parameters' tab of the 'SFDC Einstein Module Properties' dialog. It features a header with 'General' and 'Parameters' tabs. Below the header is a section titled 'Initiating of Called Script Variables' with a table. The table has two columns: 'Called Script Variable' and 'Value/Current Script Variable'. The table contains four rows of data. Below the table are 'Add', 'Remove', and 'Edit' buttons. There is a checkbox labeled 'Pass Contact variables to the called script'. Below this is a section titled 'Returned Variables' with another table. The table has two columns: 'Called Script Variable' and 'Current Script Variable'. The table contains one row of data. Below the table are 'Add', 'Remove', and 'Edit' buttons. There is a checkbox labeled 'Return Contact variables from the called script'. At the bottom right are 'Save' and 'Cancel' buttons.

Called Script Variable	Value/Current Script Variable
auth_profile_name	auth_profile
IDocument	Omni.subject
IModelId	modelID
IPredictType	"intent"

☐ Pass Contact variables to the called script

Called Script Variable	Current Script Variable
OResponse	__ExtContactFields__

☐ Return Contact variables from the called script

- **auth-prof-name:** Name of an authorization profile configured for Einstein engine in VCC.
- **IModelId:** Name of a module that is specified in SFDC Einstein. See [Einstein documentation](#).

- **IPredictType:** Type of Einstein API. Supported values:
 - sentiment
 - intent
 - vision
- **IDocument:** String to be sent to Einstein. Required for `sentiment` and `intent` types.
- **ISampleLocation:** URL for an image file. Required for `vision` only.

Select check box to pass contact variables to the called script.

- **Output Parameters:** Contains response attributes listed as KVList pairs with parameters of the model and their probabilities evaluated by SFDC Einstein.
 - For **sentiment**, the keys are strings positive, negative or neutral. The associated values are their probabilities.

Example

```
__ExtContactFields__ = { "positive": "0.8673582",  
"negative": "0.1316828", "neutral": "0.0009590242" }
```

- For **intent** the keys are intents defined during the model training. The associated values are their probabilities.

Example

```
__ExtContactFields__ = { "sales": "0.8673582",  
"support": "0.1316828", "human  
resources": "0.0009590242" }
```

- For **vision** the keys are recognized objects defined during the model training, the associated values are probabilities of the recognition.

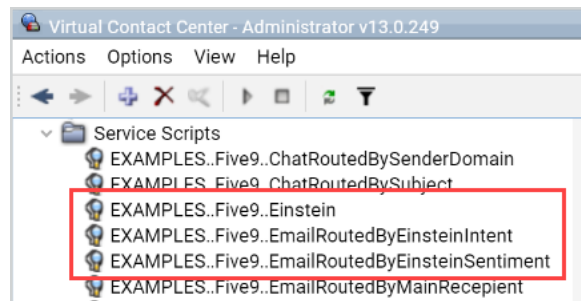
Example

```
__ExtContactFields__ = { "Beaches": "0.97554934",  
"Mountains": "0.024450686" }
```

- **Possible Errors:**
 - **91:** System errors; any problem while connecting to Einstein or parsing a response.
 - **92:** Inconsistent or unspecified input parameters.
 - **93:** **Auth_profile_name** not specified.

Example IVR Scripts with SFDC Einstein Module

To see an example as a reference before you create your own, you can open any of the example SFDC Einstein files listed under the **IVR > Service Scripts** subfolder. You can copy and rename any of the example scripts to use in your contact center routing.



Extension Transfer Module

The Extension Transfer module transfers a call to an agent after the caller enters a four-digit extension. The module plays the prompt *Please enter the extension of the party you wish to reach*. If a user account matches the extension number, the call is transferred to the agent or to voicemail if the agent is not available.

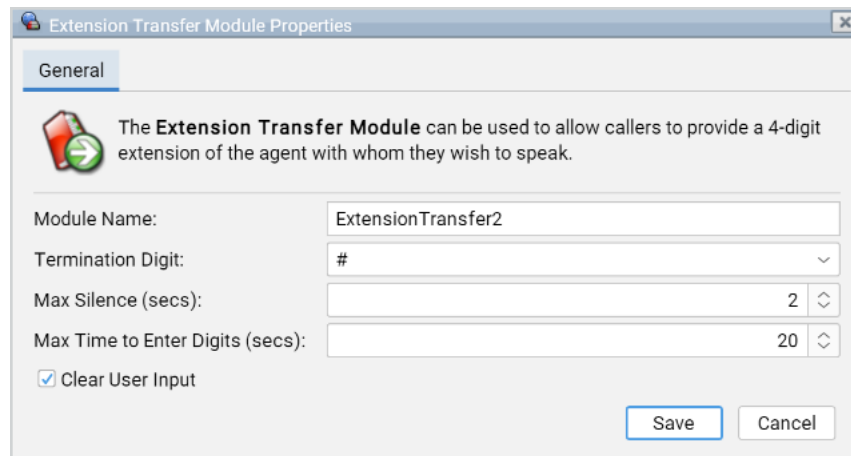
This module combines the Get Digits and two Play modules with standard files played during the module execution (input prompt) and on wrong user input.

Because this module has two connector ports, it can be used as a termination module and as a child module. If a Play module precedes the Extension Transfer module, the prompt in the Extension Transfer module is not played, which enables you to add customized instructions for entering an extension in the Play module.

This module handles invalid entries and extensions as follows:

- When the Extension Transfer is used as a termination module, this module plays the message *Extension xxxx is not available* followed by *Please enter the extension of the party you wish to reach*. The caller may enter an extension up to three times. The call is disconnected after three invalid entries.
- When the Extension Transfer is followed by another module, the IVR script proceeds to the following module.

General Tab



The screenshot shows the 'Extension Transfer Module Properties' dialog box with the 'General' tab selected. The dialog contains a description of the module's function, several input fields for configuration, and a checkbox for 'Clear User Input'. The fields are: 'Module Name' (text box with 'ExtensionTransfer2'), 'Termination Digit' (dropdown menu with '#'), 'Max Silence (secs)' (spin box with '2'), and 'Max Time to Enter Digits (secs)' (spin box with '20'). The 'Clear User Input' checkbox is checked. 'Save' and 'Cancel' buttons are at the bottom right.

Extension Transfer Module Properties

General

The **Extension Transfer Module** can be used to allow callers to provide a 4-digit extension of the agent with whom they wish to speak.

Module Name:

Termination Digit:

Max Silence (secs):

Max Time to Enter Digits (secs):

☒ Clear User Input

Save Cancel

- **Termination Digit:** Option for callers to exit the module without entering an extension by pressing the key specified.
 - Enabled: the possible options are # (pound) and * (star).
 - Disabled: the value N/A disables the option.
- **Max Silence:** Number of seconds that the module waits between digits when callers are entering the extension. The default is 2 seconds.
- **Max Time to Enter Digits:** Number of seconds that the module waits for callers to finish entering the extension. If callers exceed the time, the IVR script proceeds to the next module or repeats the current module. The default is 20 seconds.
- **Clear User Input:** Whether information entered by the caller before reaching this module is erased.
 - Enabled: delete all user input before the Extension Transfer module.
 - Disabled: do not enable if you have a prompt that provides instructions before the Extension Transfer module.

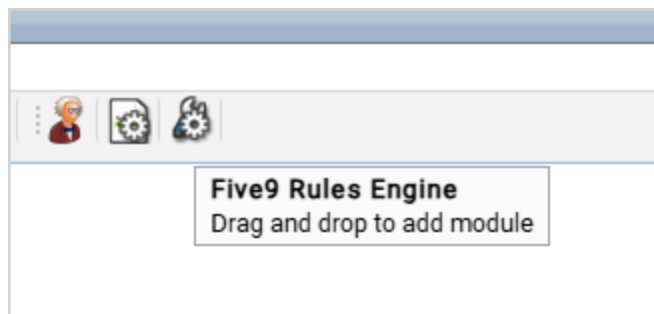
Dispositions Tab

The default disposition is Transferred to 3rd Party. For more information, see [Selecting Dispositions](#).

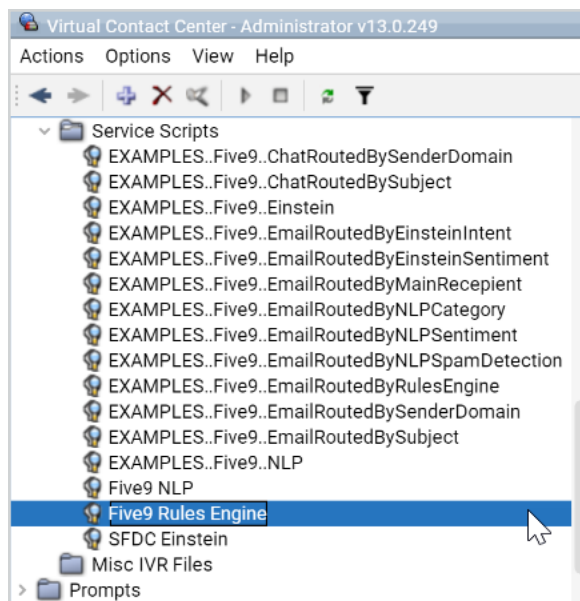
Five9 Rules Engine Module

The Five9 Rules Engine module enables you to provide more granularity to your email distribution. You can send key-value attributes as requests to the rule engine. The rule engine returns the list of attributes updated by your defined rules file. This module executes rules and updates the database with tags, attributes, and/or next best actions.

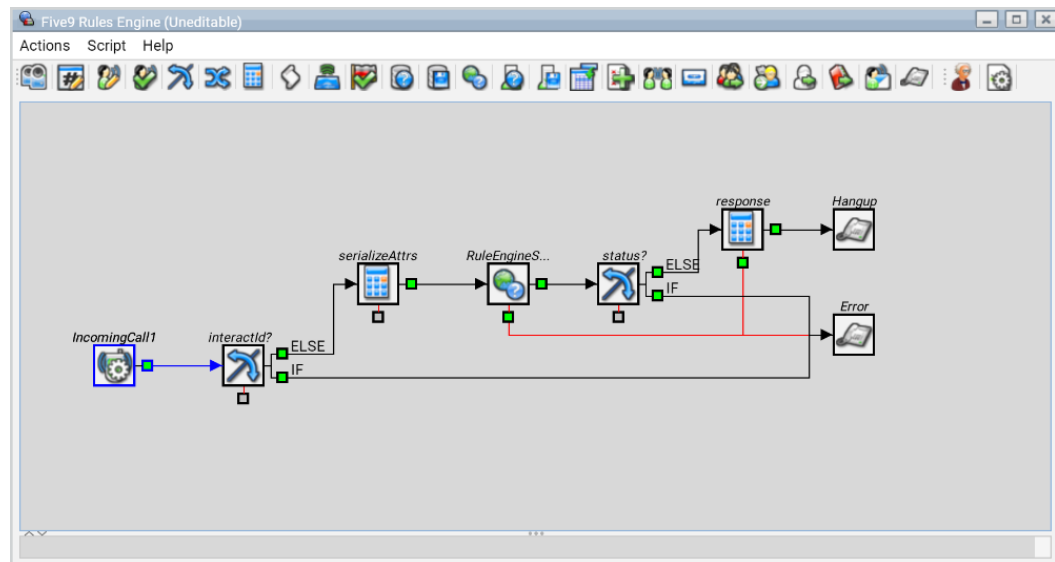
You can access the Five9 Rules Engine from the IVR Builder toolbar.



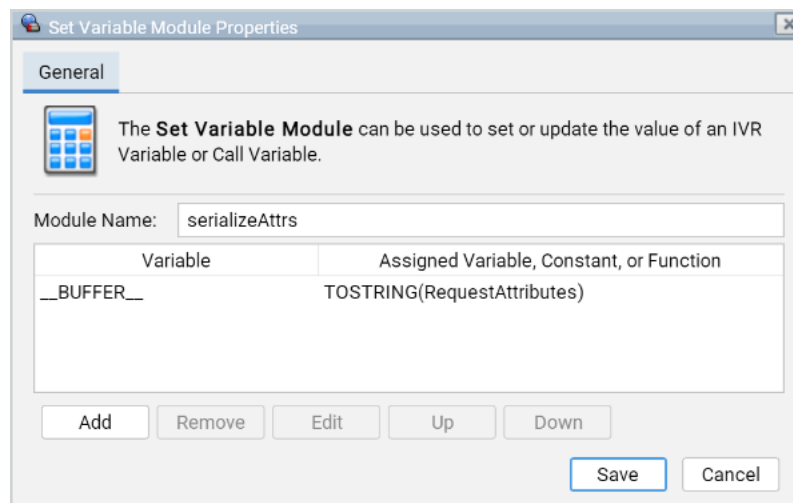
For your reference, you can access a read-only reference of the Five9 Rules Engine script at **IVR Scripts > Service Scripts > Five9 Rules Engine**.



The Five9 Rules Engine module is a pre-packaged IVR script within a module. The read-only example is shown below.



You can view each module used within the script. For example, if you open the properties for the first Set Variable module, it shows the request attributes. This value is sent to query the Five9 real-time rule engine. The response is received as a list of key-value pairs.



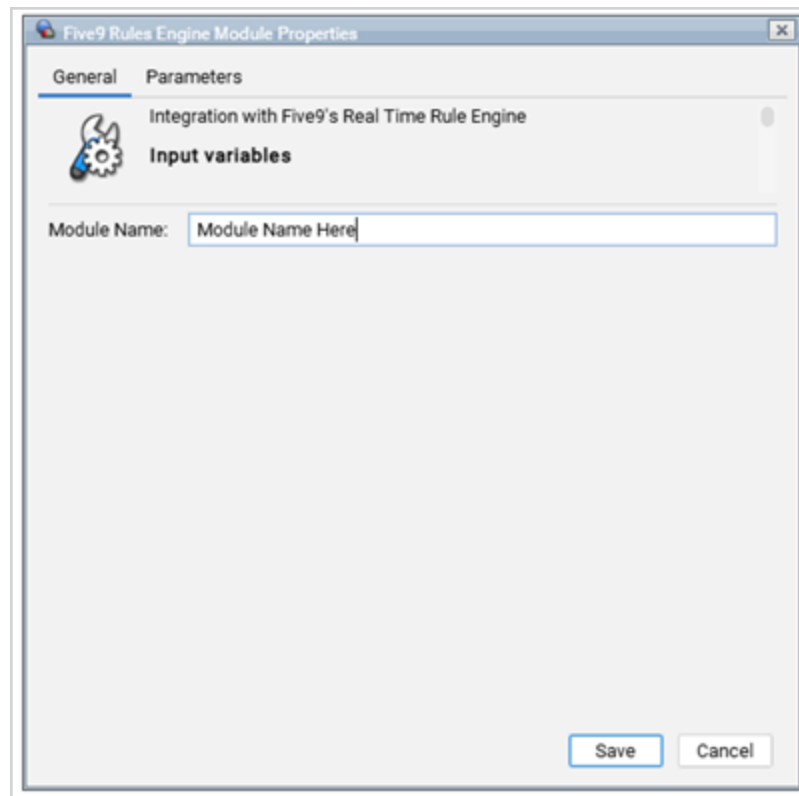
In addition to the attributes received, the rule engine may also use the body of the email to enhance routing decisions. You can use regular expressions or a Drools file. In your Five9 Rules Engine definition file, the attributes are in the **DynamicAttributes** object; the email body is located in the `getBody()` method in the `workitem` object. The rule engine also returns the list of attributes received in the request. For information about creating a Drools file, refer to the Engagement Workflow chapter of the **Five9Digital Engagement Administrator's** guide.

The following example shows a sample criteria KVList for email spam filtering.

```
{'1':'Undeliverable','2':'MAILER-DAEMON','3':'Automatic  
Reply','4':'Delivery Status Notification','5':'OPT  
OUT','6':'Instagram','7':'wine','8':'erection','9':'Online phar-  
macy','10':'Viagra','11':'Vicodin','12':'Xanax','13':'Reverses  
aging','14':'Lose weight','15':'Removes  
wrinkles','16':'Valium','17':'Cures bald-  
ness','18':'hormone','19':'Diarrhea','20':'As seen on','21':'Buy dir-  
ect','22':'Earn $','23':'Make $','24':'One time mailing','25':'Weight  
loss','26':'All nat-  
ural','27':'50%off','28':'Heute','29':'Morgen','30':'benutzen','31':'kön-  
nen','32':'gehen','33':'machen','34':'und','35':  
'das','36':'out of the office','37':'free','38':'win'}
```

Called script variables are sent and returned by using the rules engine module and Drools file.

General Tab



- **Module Name:** Name shown in IVR builder.

Parameters Tab

- **Initiating of Called Script Variables:** Contains called script with variable parameters:

The image shows a screenshot of the 'Five9 Rules Engine Module Properties' dialog box, specifically the 'Parameters' tab. The dialog has a title bar with a close button. Below the title bar, there are two tabs: 'General' and 'Parameters', with 'Parameters' being the active tab. The main content area is divided into two sections: 'Initiating of Called Script Variables' and 'Returned Variables'. Each section contains a table with two columns. The 'Initiating' table has columns 'Called Script Variable' and 'Value/Current Script Variable'. The 'Returned' table has columns 'Called Script Variable' and 'Current Script Variable'. Below each table are 'Add', 'Remove', and 'Edit' buttons. There are also two checkboxes: 'Pass Contact variables to the called script' and 'Return Contact variables from the called script'. At the bottom right, there are 'Save' and 'Cancel' buttons.

Called Script Variable	Value/Current Script Variable
------------------------	-------------------------------

Add Remove Edit

☐ Pass Contact variables to the called script

Called Script Variable	Current Script Variable
------------------------	-------------------------

Add Remove Edit

☐ Return Contact variables from the called script

Save Cancel

Important

The Five9 Rules Engine evaluates messages up to 20,000 characters in length. Messages exceeding this limit may be distributed to the default skill associated with the assigned profile.

- Response transformation field: Add regular expressions enclosed in parentheses. Regular expressions are automatically added to the RegExp Group field.

Example

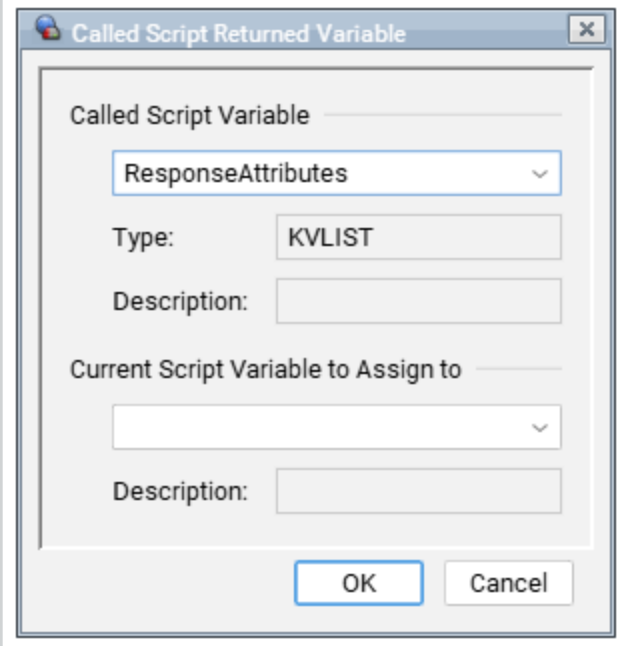
This expression matches the first instance of one or more characters.

```
<State>(.*?)</State>-<City>(.*?)</City>(.*?)
```

- Map the expressions to IVR variables in the RegExp Group field. Assign the result to a variable. In this case, the result is assigned to the `City` variable.

Enable option to pass contact variables to the called script.

- **Returned Variables:** Contains response attributes listed as key-value pairs.



The image shows a dialog box titled "Called Script Returned Variable". It contains the following fields:

- Called Script Variable:** A dropdown menu with "ResponseAttributes" selected.
- Type:** A text field containing "KVLIST".
- Description:** An empty text field.
- Current Script Variable to Assign to:** An empty dropdown menu.
- Description:** An empty text field.

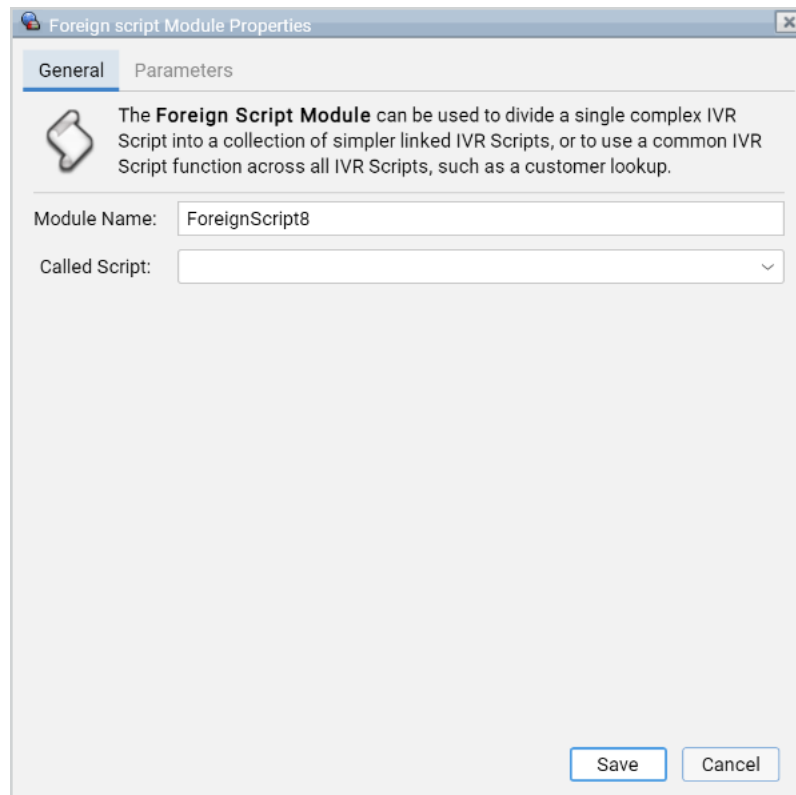
At the bottom of the dialog are "OK" and "Cancel" buttons.

Foreign Script Module

The Foreign Script module enables you to use an IVR script that contains at least one variable in the current script. The variables of the foreign script enable you to set the variables in the current script so that variables are returned in your current script. For example, you can use the same search script in multiple scripts. You can use the Foreign Script module after any module with defined variables.

Because you may not use call variables that contain sensitive information in this field, these variables do not appear in the menu even if they exist in the Call Variables folder of the VCC.

General Tab



The screenshot shows a dialog box titled "Foreign script Module Properties" with two tabs: "General" and "Parameters". The "General" tab is selected. It contains a description of the Foreign Script Module, a text field for "Module Name" with the value "ForeignScript8", and a dropdown menu for "Called Script". At the bottom right are "Save" and "Cancel" buttons.

Foreign script Module Properties

General Parameters

The **Foreign Script Module** can be used to divide a single complex IVR Script into a collection of simpler linked IVR Scripts, or to use a common IVR Script function across all IVR Scripts, such as a customer lookup.

Module Name: ForeignScript8

Called Script: [Dropdown Menu]

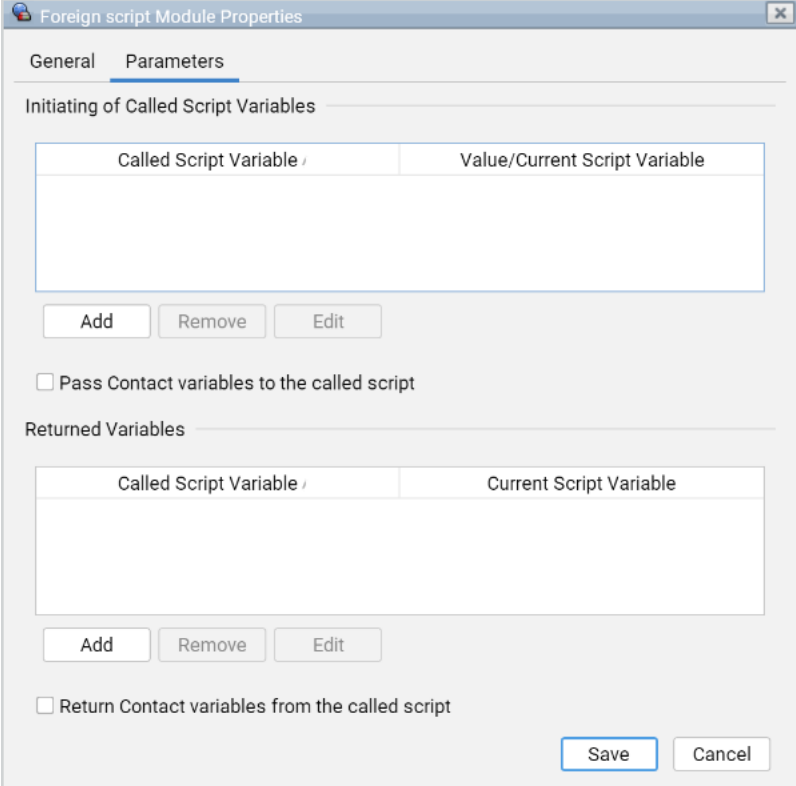
Save Cancel

- **Called Script:** List of your scripts. When you select a script, the Parameters tab is enabled, but the menus are disabled unless the foreign script contains variables.

Parameters Tab

Because you may not use call variables that contain sensitive information in this field, these variables do not appear in the menu even if they exist in the Call Variables folder of the VCC.

- **Initiating Called Script Variables:** Variables to pass to the foreign script.



The dialog box is titled "Foreign script Module Properties" and has two tabs: "General" and "Parameters". The "Parameters" tab is selected. It contains two sections: "Initiating of Called Script Variables" and "Returned Variables".

Initiating of Called Script Variables

Called Script Variable	Value/Current Script Variable
------------------------	-------------------------------

Buttons: Add, Remove, Edit

☐ Pass Contact variables to the called script

Returned Variables

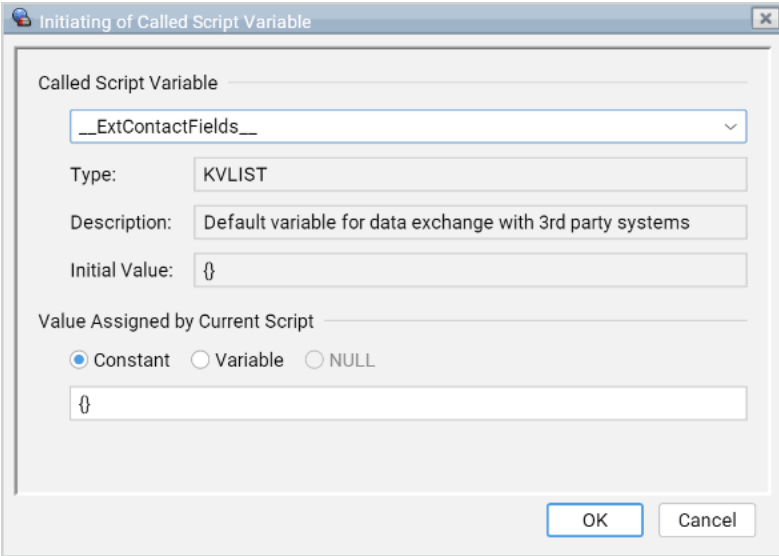
Called Script Variable	Current Script Variable
------------------------	-------------------------

Buttons: Add, Remove, Edit

☐ Return Contact variables from the called script

Buttons: Save, Cancel

To set the variables of the foreign script to values of the current script, click **Add**.



The dialog box is titled "Initiating of Called Script Variable". It contains the following fields and options:

Called Script Variable

__ExtContactFields__

Type: KVLIST

Description: Default variable for data exchange with 3rd party systems

Initial Value: {}

Value Assigned by Current Script

☒ Constant ☐ Variable ☐ NULL

{}

Buttons: OK, Cancel

- Called Script Variable: List of variables in the foreign script, data type of the variables, and optional description and initial value.
- Value Assigned by Current Script: Value to assign to the variable:
 - Constant: Single value that you specify.
 - Variable: >List of your user, IVR, and user CAV variables. When you select a variable, the description is automatically populated.
 - NULL: Empty field.
- **Pass contact variables to the called script:** Whether to pass contact variables to the foreign script:
 - Enabled: Contact variables are passed.
 - Disabled: Contact variables are not passed.
- **Returned Variables:** Variables that store the returned data:
 - Called Script Variable: List of variables in the foreign script, data type of the variables, and optional description and initial value.
 - Current Script Variable to Assign to: Variable that stores the returned data. The list contains your user, IVR, and user CAV variables. When you select a variable, the description is automatically populated.
- **Return contact variables from the called script:** Whether to accept contact variables from the foreign script:
 - Enabled: Contact variables are returned.
 - Disabled: Contact variables are not returned.

Get Digits Module

The Get Digits module enables you to route the call by using digits entered by the caller into a variable. When the module receives the correct digits, the IVR script proceeds to the next module. The result of this module can be assigned to a string or integer variable.

Important

When you use custom variables that contain sensitive data, the values are replaced by#### in the VCC server logs and cannot be saved in the database.

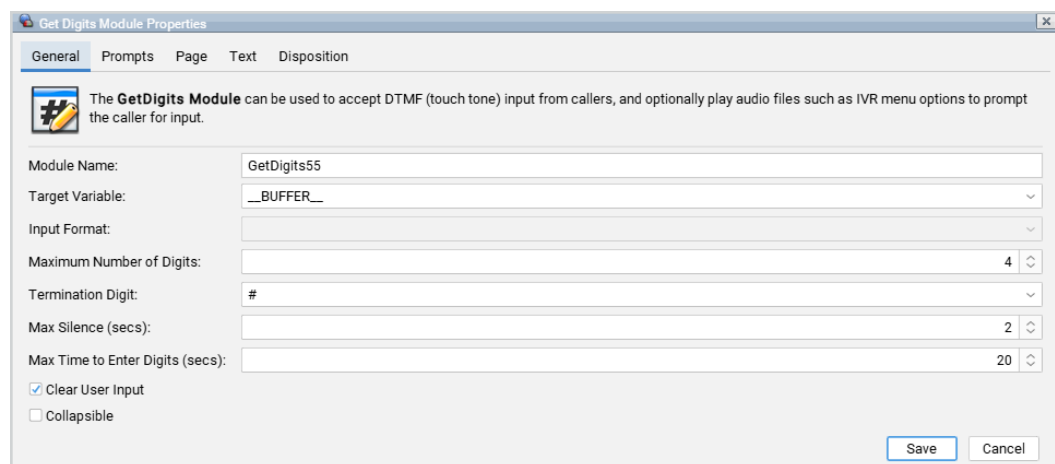
If your domain is not enabled for the E.164 format for international phone numbers, agents must enter 10 digits to conform to the [North American Numbering Plan](#). If your domain is enabled for the E.164 format, agents must enter phone numbers for all calls according to your configuration: in your national format or up to 15 digits, starting with the plus (+) sign and the country code. If you do not specify a format, the VCC uses the E.164 format for all numbers. For more information, see the [Basic Configuration Administrator's Guide](#).

When using Visual IVR scripts, you may also add images to pages and menus to enhance your brand or to describe a product or service. However, you cannot add links to images. For more information, see [Page Tab](#).

For example uses, see [Creating an IVR Script](#), [Creating an IVR Loop](#), and [Using an Agent Extension in an IVR](#).

[General Tab](#)
[Prompts Tab](#)
[Page Tab](#)
[Text Tab](#)
[Dispositions Tab](#)

General Tab



The screenshot shows the 'Get Digits Module Properties' dialog box with the 'General' tab selected. The dialog has a title bar with a close button (X). Below the title bar are tabs for 'General', 'Prompts', 'Page', 'Text', and 'Disposition'. The 'General' tab contains a description: 'The GetDigits Module can be used to accept DTMF (touch tone) input from callers, and optionally play audio files such as IVR menu options to prompt the caller for input.' Below this are several configuration fields: 'Module Name' (text box with 'GetDigits55'), 'Target Variable' (dropdown menu with '__BUFFER__'), 'Input Format' (dropdown menu), 'Maximum Number of Digits' (spin box with '4'), 'Termination Digit' (dropdown menu with '#'), 'Max Silence (secs)' (spin box with '2'), and 'Max Time to Enter Digits (secs)' (spin box with '20'). At the bottom left are two checkboxes: 'Clear User Input' (checked) and 'Collapsible' (unchecked). At the bottom right are 'Save' and 'Cancel' buttons.

- **Module Name:** Name that represents the role of the module in the IVR script. In Visual IVR scripts, the name is converted to text that appears in the heading of the IVR window.

- **Target Variable:** List of user, contact, IVR, and user CAV variables that can store the caller's input. You can also create a set of custom variables in **Script > Variables**.
- **Input Format:** Format template for the caller's input. This table contains the input format options for entering data:

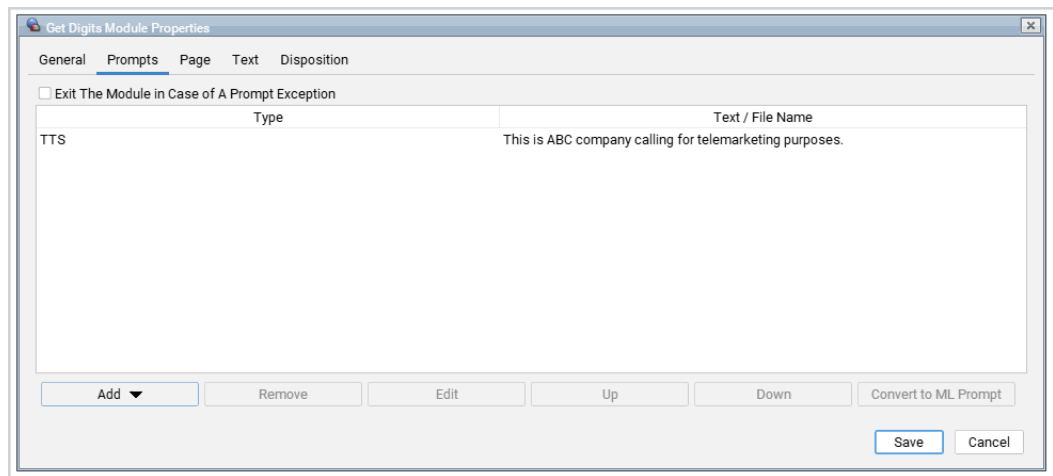
Variable Data Type	User Input
Date	MMDDYY
	MMDDYYYY
	DDMMYYYY
Time	HHMM
	HHMMP
Currency	\$...\$CC
	\$...\$

- **Maximum Number of Digits:** Number of digits allowed before proceeding to the next module.
- **Termination Digit:** Option for callers to exit the module and proceed to the next module by pressing the key specified.
 - Enabled: the possible options are # (pound) and * (star).
 - Disabled: the value N/A disables the option.
- **Max Silence:** Number of seconds that the module waits between digits. The default is 2 seconds. Does not apply if the maximum number of digits is set to 1.
- **Max Time to Enter Digits:** Number of seconds that the module waits for callers to finish entering digits. If callers exceed the time, the script proceeds to the next module or repeats the current module. The default is 20 seconds.
- **Clear User Input:** Whether to erase the information entered by the caller before reaching this module.
 - Enabled: Delete all user input before this module.
 - Disabled: Do not enable if you have a prompt that provides instructions, such as pressing a key (termination digit) to exit the queue, before this module. Otherwise you erase the caller's input in the Play module.

Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). Each prompt can be interruptible.

In Visual IVR scripts, TTS prompts are converted to text that appears in the body of the IVR window.

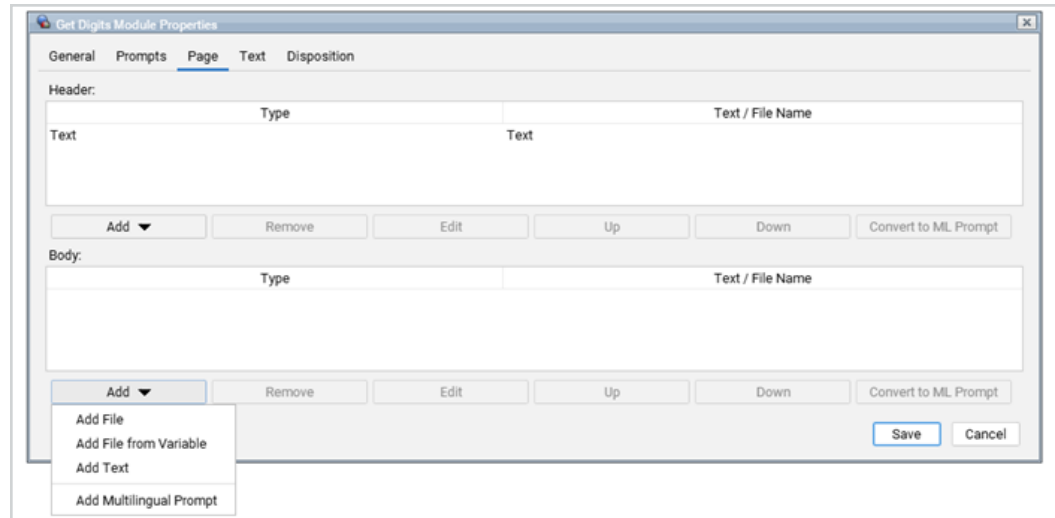


- **Exit the module in case of a prompt exception:** Whether to proceed to the next module if a prompt error occurs:
 - Enabled: Proceed to the next module.
 - Disabled: Exit the script.

Page Tab

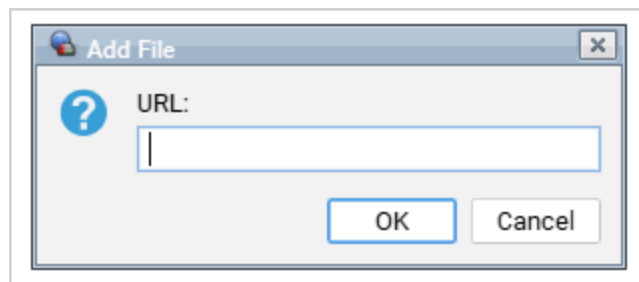
This tab contains the list of Visual IVR prompts that are shown visually in the customer's Web browser when a IVR script is executed.

- If this tab contains prompts, these are shown whereas prompts in the Prompts tab are not.
- If this tab does not contain prompts, the prompts in the Prompts tab are shown.

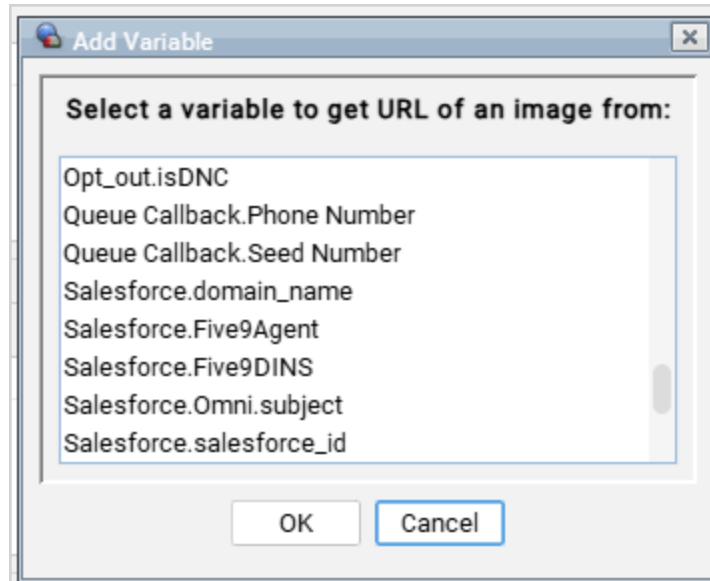


This tab enables you to add different types of prompts:

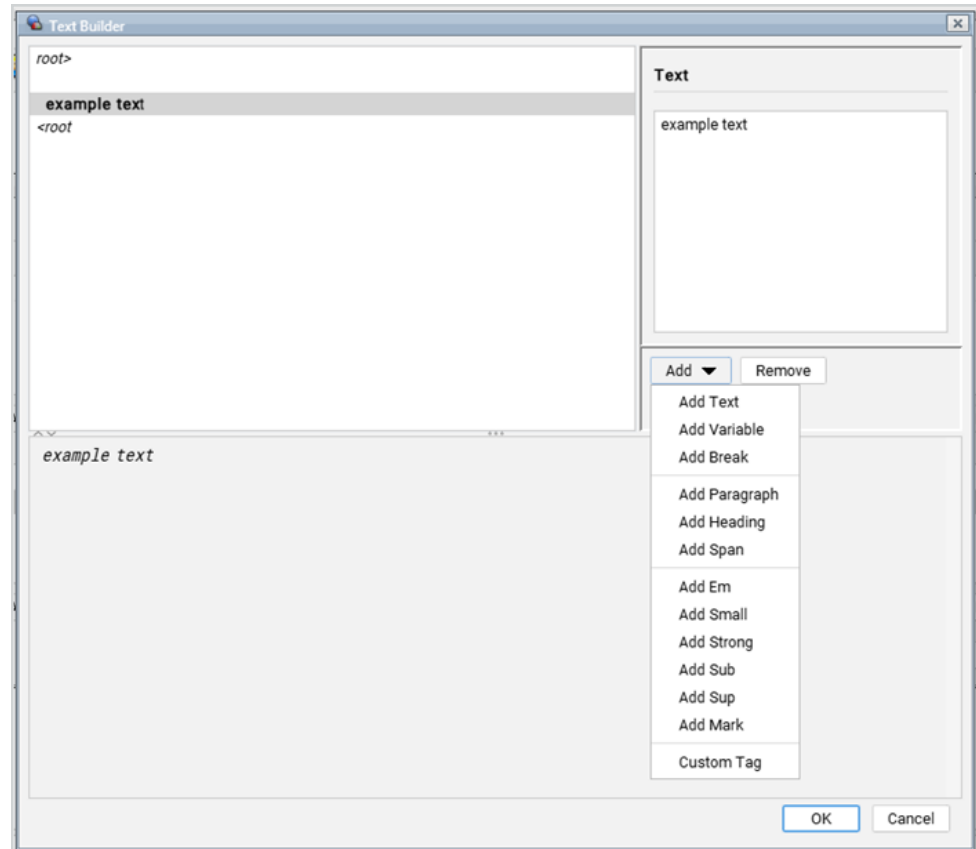
- **File:** URL of an image.



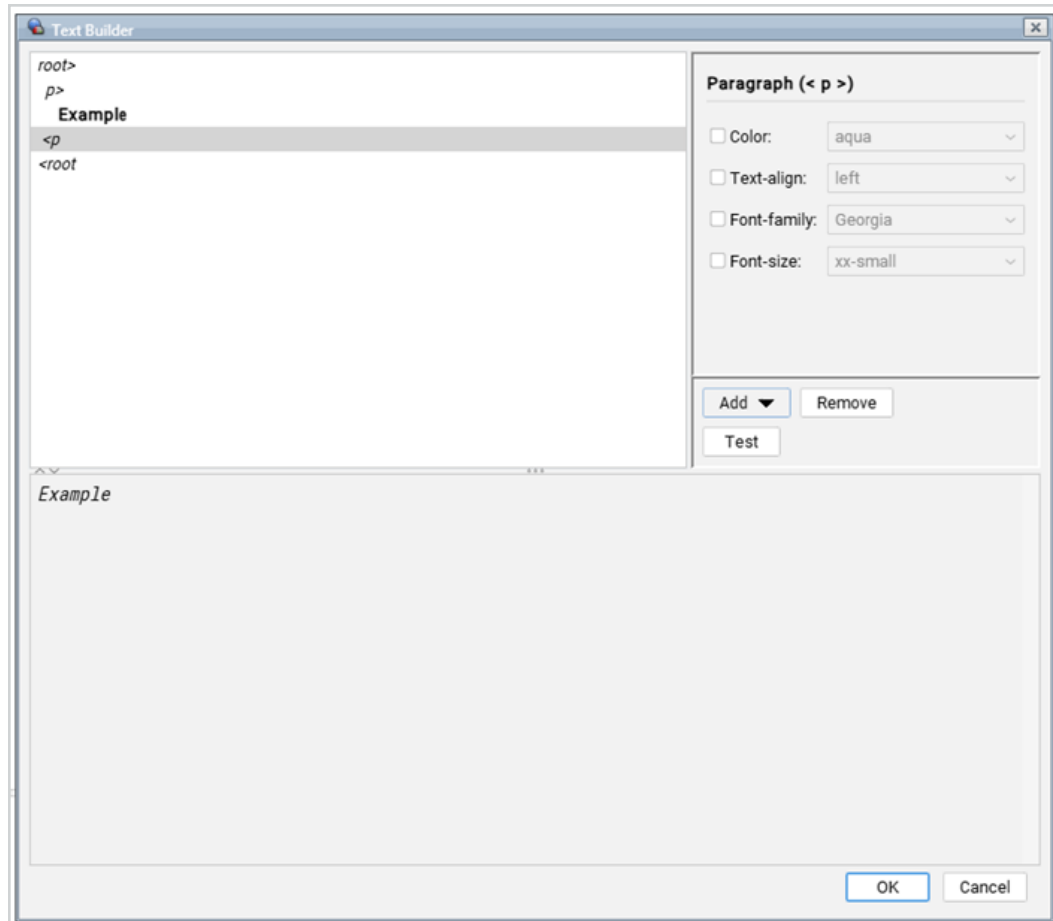
- **Add File from Variable:** Variable that contains the URL of an image.



- **Text:** Formatted text that can contain variables. You cannot add images. If you select a variable that contains the URL of an image, the URL is displayed.
- **Custom Tag:** For more information, see [Using the Visual IVR URL](#).

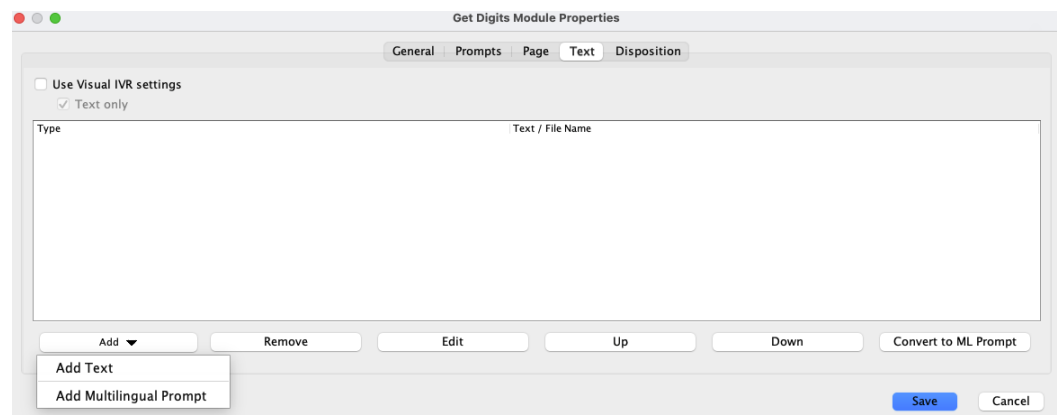


The text editor enables you to create HTML text that contains formatted text and variables. You can customize the text with bold, color, and emphasis. Unlike breaks in the TTS builder, which are associated with speech, breaks in this text editor add a line or horizontal break. You can use the custom tag element to use specific HTML elements. The Test button enables you to preview the HTML text.



Text Tab

This tab enables configuration options for chat, SMS, email, and integration channels.



- **Use Visual IVR Settings:**
 - Enabled: digital engagement interactions receive the data configured on the **Page** tab.
Text only: If enabled, the XML/HTML are removed from the data configured on the Page tab.
 - Disabled:
 - The Text only box is disabled.
 - **Add Text:** Text and variables can be added.
 - **Add Multilingual Prompt:** Only multilingual prompts with the type TEXT may be used here.

Dispositions Tab

The default disposition is Caller Disconnected. For more information, see [Selecting Dispositions](#).

If/Else Module

The If/Else module enables you to route the script according to a condition, such as a logical expression that uses variables and constants. The module has two branches:

- If the condition is true, the flow uses the If branch.
- If the condition is false, the flow uses the Else branch.

If you need more than two branches, use the Case module. When the condition is boolean, you need to create only the If branch. Otherwise, the condition exits the module by the Else branch. For examples, see [Creating an IVR Script](#) and [Creating an IVR Loop](#).

General Tab

The screenshot shows the 'If/Else Module Properties' dialog box with the 'General' tab selected. The dialog contains a description of the module, a 'Module Name' field with the value 'IfElse44', a 'Condition Grouping' dropdown set to 'All', and an 'Expression' field. Below these is a table for defining conditions. The table has five columns: 'Criteria #', 'Variable', 'Operator', 'Variable', and 'Value/Variable'. At the bottom of the dialog are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', 'Save', and 'Cancel'.

If/Else Module Properties

General

The **If/Else Module** can be used to select one of two branches in the IVR Script based on a logical expression such as the value of a variable or input previously provided by a caller.

Module Name: IfElse44

Condition Grouping: All

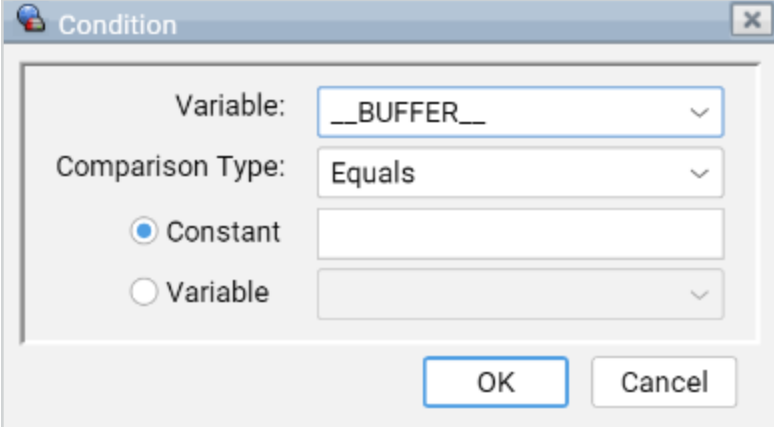
Expression:

Criteria #	Variable	Operator	Variable	Value/Variable
------------	----------	----------	----------	----------------

Add Remove Edit Up Down Save Cancel

- **Condition Grouping:** Method used to join conditions:
 - Any: Any defined criteria are valid. The expression 1 OR 2 is entered automatically in the Expression field.
 - All: All defined criteria must be valid. The expression 1 AND 2 is entered automatically in the Expression field.
 - Custom: Custom expression that describes how to use the criteria. This option enables the Expression field.
- **Expression:** Logical expression that describes how to compare the conditions. If you have a complex expression, such as 1 AND 2 OR 3 AND 4, it is evaluated as (1 AND 2) OR (3 AND 4) because AND is always evaluated before OR.
- **Condition table:** Numbered list of conditions that you can add, modify, delete, and arrange as needed. Criteria # represents the condition in the expression field.

- **New condition window:**



- Variable: Left variable to be compared to a constant or another variable. List of your user, contact, system, agent, call, IVR, or user CAV variables.
- Comparison Type: The available operators depend on the data type of the left variable.

Left Variable	Comparison Types	Right Variable
INTEGER	Equals	INTEGER
NUMBER	Less than	NUMBER
CURRENCY	More than	CURRENCY
STRING	Equals	STRING
	Like	Like supports wildcards % and _.
	RegExpr	
DATE or TIME	Equals	DATE or TIME in the specified internal format
	Less than	
	More than	
For all variables	IsNull	Empty field

- Right variable to be compared:
 - Constant: Specific value. The only boolean values accepted are lowercase `true` and `false`.
 - Variable: List of your user, contact, system, agent, call, IVR, or user CAV variables.

Input Module

The Input module analyzes the words spoken by callers or DTMF digits. Phrase recognition is based on predefined grammar. The meaning of the input is assigned to a variable.

[Digital Channels](#)

[Languages](#)

[Properties](#)

[General Tab](#)

[Prompts Tab](#)

[Events Tab](#)

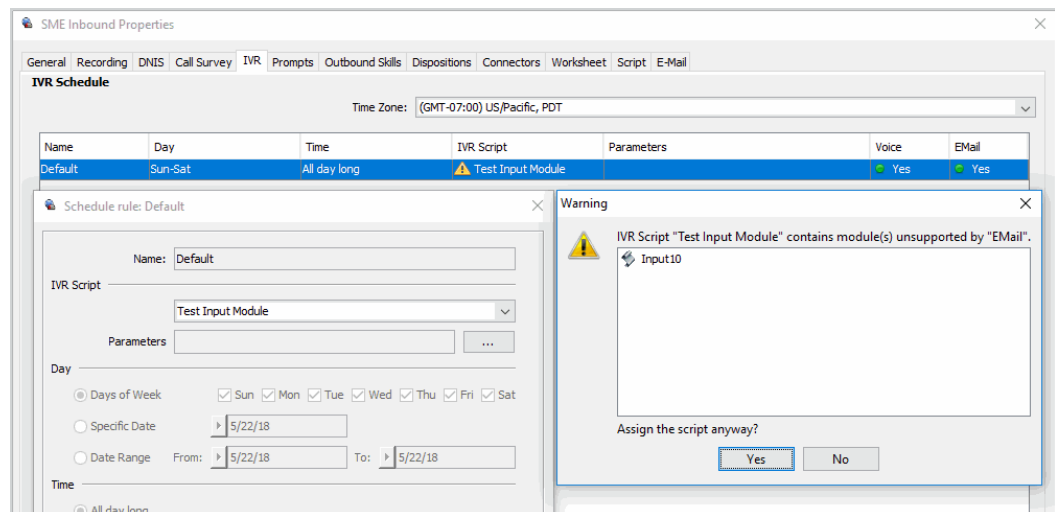
[Confirmation Tab](#)

[Page Tab](#)

[Text Tab](#)

Digital Channels

The Input module is not compatible with digital channels. If your domain is enabled for digital channels, you see a warning when you use IVR scripts that contain this module in inbound campaigns. If you choose to use the script, a warning icon is displayed in the schedule description. The IVR schedule still works, but these incompatible modules are skipped.



Languages

You may have the option to use a language other than English.

- If your domain is not enabled for the E.164 format for international phone numbers, agents must conform to the [North American Numbering Plan](#) by entering 10 digits.
- If your domain is enabled for the E.164 format, agents enter phone numbers according to your configuration: in your national format or up to 15 digits, starting with the plus (+) sign and the country code. If you do not specify a format, the E.164 format is used for all numbers.

Properties

This section describes the options available in this module.

[General Tab](#)
[Prompts Tab](#)
[Events Tab](#)
[Confirmation Tab](#)
[Page Tab](#)
[Text Tab](#)

General Tab

The **Input Module** can be used to analyze spoken caller input or DTMF for short phrases based on pre-defined grammars for numbers, credit cards, zip codes, social security numbers, phone numbers, dates, time, and Boolean (Yes/No) phrases, so that the call will follow the appropriate branch in the IVR application

Module Name:

Variable:

Grammar:

Properties:

Appl	Property	Value
☒	language	en-US
☒	y	1
☒	n	2

Additional Returned Values:

Return Key	Variable to Save Value
SWI_literal	<Ignore>
Confidence Level	<Ignore>

Max Attempts:

DTMF Help Button:

☐ Record User's Input as Audio File

☐ Collapsible

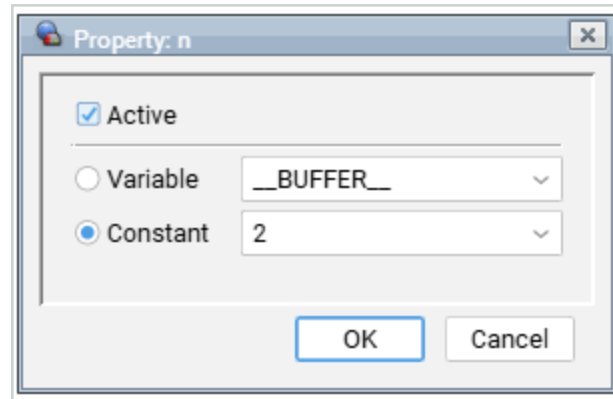
- **Module Name:** Name of the module in the script. In Visual IVR scripts, the name is displayed in the heading of the IVR window.
- **Variable:** Value saved as the variable (list of user CAV variables).
- **Grammar:** Speech recognition grammar is information that you can collect to obtain the meaning for the selected variable. Each type may have configurable properties and additional returned values. Five9 supports 2 active grammars: speech and DTMF. For detailed information, see [Speech Recognition Grammars](#).
 - **alphanumeric:** Values allow entry of string formatted the same way as the input file. Non-alphanumeric characters (dash, space, and so on) are permitted but ignored in the result. Applies only to string variables.
 - **boolean:** Yes or no represented by y and n.
 - **ccexpdate:** Card expiration date.
 - **creditcard:** Credit card number that follows the ISO numbering scheme.
 - **currency:** Dollars and cents. Accepts a string in this format:
 <currency><unit amount><dot><subunit amount>.

Spoken examples: ten dollars, ten dollars and fifteen cents, and ten fifteen. For more information, see [currency](#).

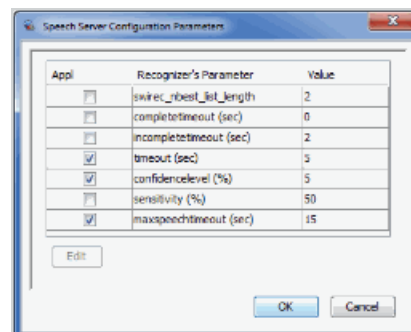
- [date](#): Date spoken in any of the formats accepted by the system.
- [digits](#): Continuously spoken string of up to 20 digits.
- **externalgrammar**: Available when a STRING variable is selected as grammar source in the Input module. Used with the ruleref URL field, which can be set to the URL of the grammar text file. The ruleref URL field is used to set the URL of the grammar text file.
 - **ruledef uri**: DTMF for active grammar support. Must be enabled to activate **rule id** and **return var** for DTMF active grammar support.
 - **rule id**: the root rule id in grammar. Accepts DTMF for active grammar support.
 - **return var**: name of the variable in the grammar file that will be returned and assigned to the selected variable. Accepts DTMF for active grammar support.
 - **ruleref uri**: explicit local rule reference.
 - **>ruleref uri DTMF**: explicit local DTMF reference. To enable the next two options, set this to **ON**.
 - **rule id DTMF**: DTMF-specific rule id.
 - **return var DTMF**: DTMF-specific return var.
 - In the visual IVR, the arbitrary grammar fields are represented as entry fields (up to 255 Characters).

Only **SWI_Literal** and **Confidence Level** are available as additional variables.

- [number](#): Spoken quantities such as ten, one hundred and forty, five hundred sixty one point five, and negative five.
 - [phone](#): 7- and 10-digit North American phone numbers and three-digit phone numbers ending in 11, such as 911. An optional 1 can be placed before 7- or 10-digit numbers.
 - [socialsecurity](#): 9-digit U.S. social security numbers.
 - [time](#): Spoken time of day.
 - [zipcode](#): United States ZIP codes in five- or nine-digit format.
- **Properties**: Properties of the type of information selected in the Grammar menu:



- Appl: Whether the property is enabled.
 - Property: Name of the property.
 - Value: Value of the property. The language property cannot be modified. You can use a constant or a call variable.
- **Advanced:** Visible if your domain is enabled for speech recognition. Opens the **Speech Server Configuration Parameters** window.



- `swirec_nbest_list_length`: Maximum number of versions of the utterance returned by the recognizer. The default is 2. The range is 0 to 999.
- `completetimeout`: Duration of the silence after the user's speech before the speech is accepted or a nomatch event is returned. Used when the speech completely matches an active grammar and no further words can be spoken. The default is 0 milliseconds.
- `incompletetimeout`: Duration of silence after the caller's words to conclude that the caller is finished speaking. The default is 1500 milliseconds.
- `timeout`: Enabled by default. Number of milliseconds to wait for speech after a prompt ends. The default is 7000.

- `confidencelevel`: Enabled by default. Percentage of confidence that the spoken text is recognized. Words that score below this value are rejected. The default is 5%.
 - `sensitivity`: Sensitivity of the speech detector. The default is 50%.
 - High percentage: More sensitive, which is better for quiet speech, but the risk that background noise is detected as speech increases.
 - Low percentage: Less sensitive, but may require that some callers speak more loudly.
 - `maxspeechtimeout`: Enabled by default. Maximum duration in milliseconds of the speech collected from users. The default is 15000. No upper limit. No timeout = -1.
- **Additional Returned Values**: After recognizing the expression, the meaning is stored in the variable selected in the Properties menu.

Depend on the selected grammar, other keys may be returned:

- **Return Key**: Information specific to the grammar, such as the text that was recognized (`SWI_Literal`) or speech recognition confidence level.
- **Max Attempts**: Number of times to repeat a prompt when a speech recognition error occurs, such as no match. The default is 3 times. `NoInput` and `NoMatch` events are considered input attempts.
- **DTMF Help Button**: DTMF key to use for the caller to request help.
- **Record User's Input as Audio File**: Whether to save the voice input as an audio file that is available in via reports. See also [Using Recorded Files](#).

IVR recordings are delivered by FTP singly or in batches. The naming convention of the files sent to the FTP server depends on whether you have checked the box **Set Name of Recording to Session ID During Transfer** for the campaign:

- Enabled: IVR recording file names have this format:
`<session ID> @ <time of call><module name>.wav`
- Disabled: IVR recording file names have this format:
`<phone number> by <agent name> @ <time of call><module name>.wav`
 - `<phone number>`: ANI (inbound), DNIS (outbound).
 - `<agent name>`: IVR recordings value is `nobody`.
 - `<time of call>`: Time that the call started.
 - `<session ID>`: Session ID of the call.
 - `<module name>`: IVR module that recorded the call.

- **Input Collapsible:** Visual IVR feature that applies to all types of prompts. Whether to concatenate all the prompts in the module whether they are played (standard prompts) or displayed (Visual IVR prompts) to callers.
 - Enabled: Concatenate prompts. Callers see at once all the prompts in the same window.
 - Disabled: Do not concatenate prompts. Callers hear or see each prompt after performing the requested action.

Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence and indicate how many times each can be used. Between prompts, you can add brief pauses (milliseconds).

- **Interruptible:** If enabled, the module stops playing the prompt as soon as the requested operation is completed.
- **Count:** Number of times that the prompt can be repeated.

In Visual IVR scripts, TTS prompts are converted to text that appears in the body of the IVR window.

Events Tab

- **Event:** List of event handling options. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in a specific sequence.

The screenshot shows the 'Input Module Properties' dialog box with the 'Events' tab selected. The dialog has a title bar with a close button. Below the title bar is a tabbed interface with tabs for 'General', 'Prompts', 'Events', 'Confirmation', 'Page', 'Text', and 'Disposition'. The 'Events' tab contains a table with the following data:

Event	Count	Event's Prompt	Interrupt.	Action
NoMatch	1	NoMatchPrompt	☐	continue
NoInput	1	NoInputPrompt	☐	reprompt

Below the table are three buttons: 'Add', 'Remove', and 'Edit'. At the bottom of the dialog, there are two spinners: 'Max Time to Enter (secs):' set to 15 and 'No Input Timeout (secs):' set to 5. At the very bottom right are 'Save' and 'Cancel' buttons.

Event: NoInput

Count: 1

Action: continue

Prompt

☐ Interruptible

Type	Text / File Name
------	------------------

Add ▼ Remove

Edit Up

Down Convert to ML Prompt

OK Cancel

List of possible events:

- NoMatch: Generated if the speech recognition engine cannot find a match in the grammar.
- NoInput: Generated if the speech recognition engine does not hear any caller input before the time-out.
- Help: Generated if the caller presses the Help button.
- **Count:** Number of possible instances (1–13) for an action, which enables you to handle differently multiple errors of the same type. The count is increased regardless of the event.

Example

NoMatch event: The module repeats the prompt to the caller. If the number of attempts is exceeded, the exception NOT_ENTERED occurs, and the script proceeds using the No Match branch to the next module.

- **Action:** Action taken when the specified event occurs.
 - Reprompt: Repeat the prompt listed in the Prompts tab.
 - Continue: Proceed to next attempt without playing the prompt listed in Prompts tab.
 - Exit: Exit by the error branch at the bottom of the module. This event can occur even if the maximum number of attempts has not been reached. The error branch must be linked to subsequent IVR modules to handle the error.
- **Prompt:** List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). Each prompt can be interruptible.
- **Max Time to Enter:** Number of seconds that the module waits for callers to finish entering the information. If callers exceed the time, the IVR script proceeds to the next module or repeats the current module. The default is 15 seconds.
- **No Input Timeout:** Number of seconds allowed for the caller to provide no input. After the time-out, the NoInput event occurs.
- **Recognition Confidence Threshold:** Percentage confidence level that the system can recognize the caller's input, which can depend on the caller's accent, the quality of the phone connection. This threshold indicates when to consider the speech recognition successful or invalid. The default is 5%. If you set the threshold too high, the number of no matches may be artificially too high.

Confirmation Tab

Prompt used to verify that the system recognized the speech correctly. If the caller confirms the input, the script proceeds to the next module. Otherwise, the prompts are repeated.

- **Required:** Confirmation prompt is played after each caller's input.
- **If Confidence Level Less Than:** Confirmation prompt played only if the confidence level is low.
- **Not Required:** Confirmation prompt disabled.

The screenshot shows the 'Input Module Properties' dialog box with the 'Confirmation' tab selected. The 'Confirmation' section has three radio buttons: 'Required' (selected), 'If Confidence Level Less Than' (with a value of 75), and 'Not Required'. The 'Confirmation Dialog' section has two spinners: 'Max Attempts to Confirm' (set to 3) and 'No Input Timeout (secs)' (set to 3). The 'Prompt' section has a text field containing 'ConfirmPromptWithoutVSR' and an 'Edit' button, with a checked 'Interruptible' checkbox below it. The 'Events' section contains a table with three columns: 'Event', 'Event's Prompt', and 'Action'.

Event	Event's Prompt	Action
NoMatch	NoMatchPrompt	reprompt
NoInput	NoInputPrompt	reprompt

At the bottom of the dialog are 'Edit', 'Save', and 'Cancel' buttons.

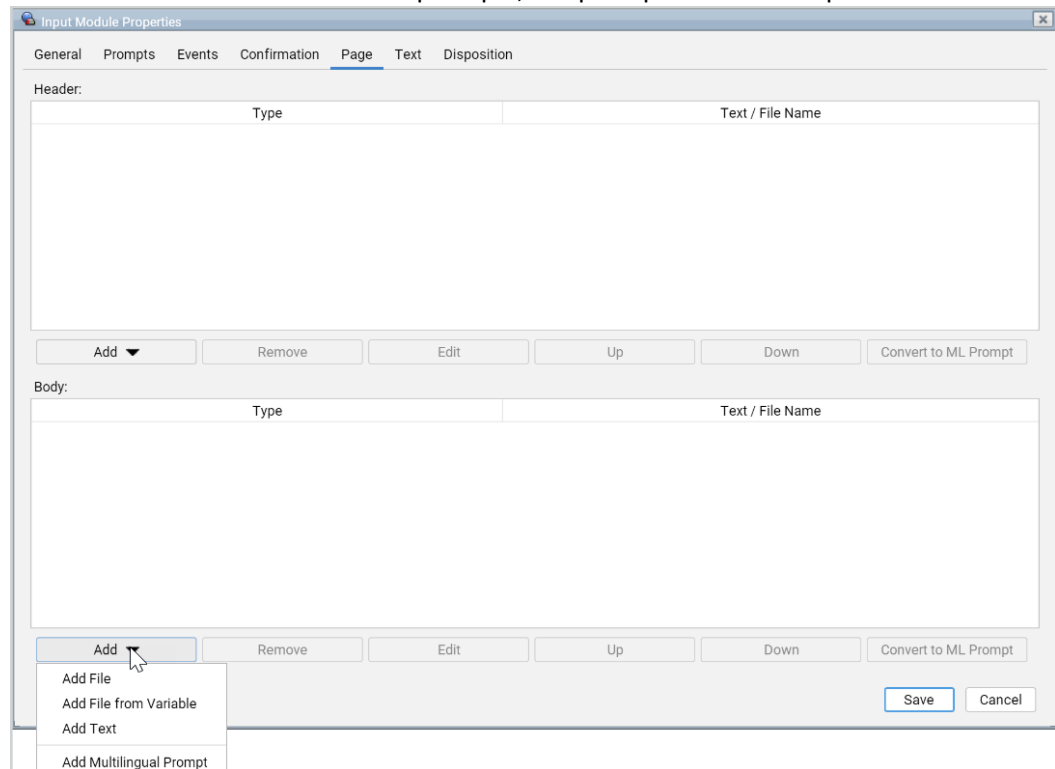
- **Max Attempts to Confirm:** Maximum number of times allowed to enter the confirmation answer. For example, the caller may say 1 or presses 1. If the number of attempts is exceeded, the exception NOT_ENTERED occurs, and the script proceeds to the next module.
- **No Input Timeout:** Number of seconds allowed for the caller to provide no input. After the time-out, the NoInput event occurs.
- **Prompt:** Prompt that describes the confirmation options. The local module variable \$SWI_LITERAL is used to repeat the caller's input as it was recognized.
- **Events:** List of possible error events:
 - NoMatch: Speech recognition feature cannot find a match in the grammar.
 - NoInput: Speech recognition feature cannot hear caller input before the time-out.
 - Help: Generated if the caller presses the Help button.

Page Tab

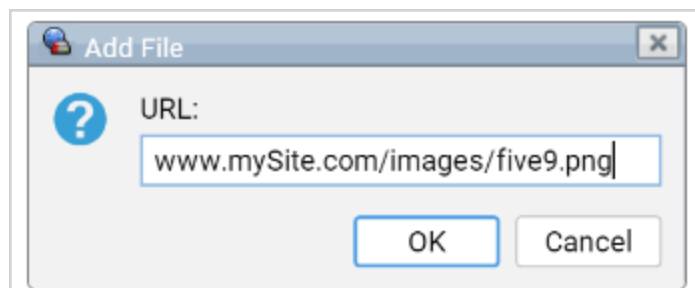
This tab contains the list of Visual IVR prompts that are shown in the customer's browser when a IVR script is executed. You may add images to pages and menus to

enhance your brand or to describe a product or service, but you cannot add links to images. Prompts are shown as follows:

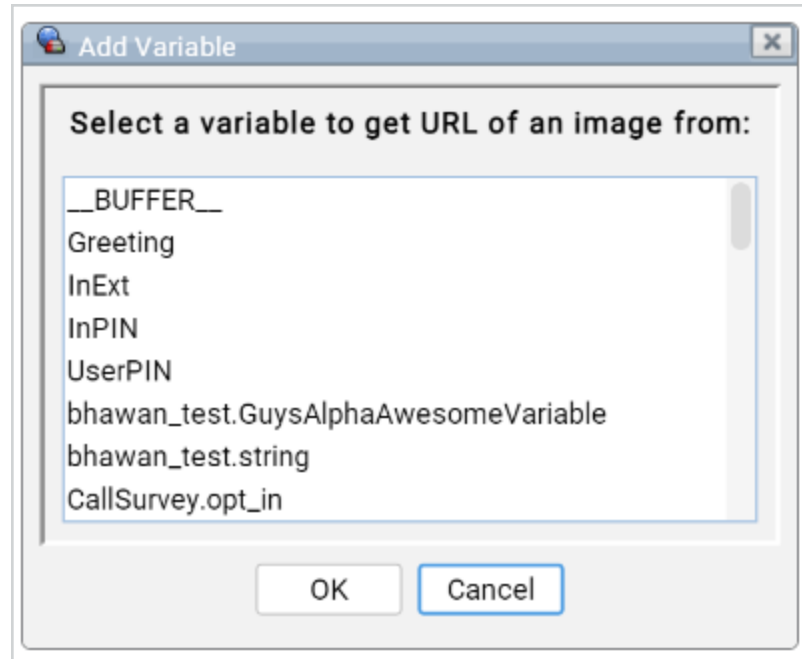
- If this tab contains prompts, these are shown whereas prompts in the Prompts tab are not.
- If this tab does not contain prompts, the prompts in the Prompts tab are shown.



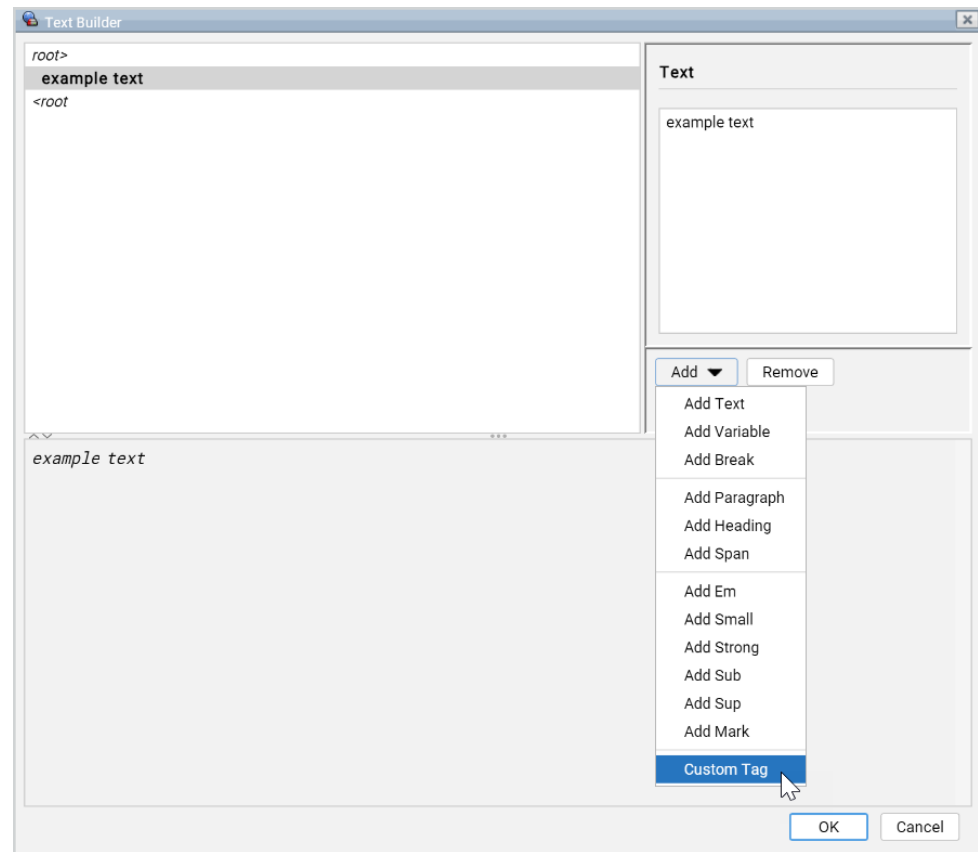
- **File:** URL of an image.



- **Variable:** Variable that contains the URL of an image.



- **Text:** Formatted text that can contain variables. You cannot add images. If you select a variable that contains the URL of an image, the URL is displayed.
- **Custom Tag:** See detailed usage information in [Using the Visual IVR URL](#).



You can create HTML that contains formatted text and variables, and you can customize the text with bold, color, and emphasis. Unlike breaks in the TTS builder, which are associated with speech, breaks in this text editor add a line or horizontal break. Click Test to preview the HTML text.



Text Tab

This tab enables configuration options for chat, SMS, email, and integration channels.

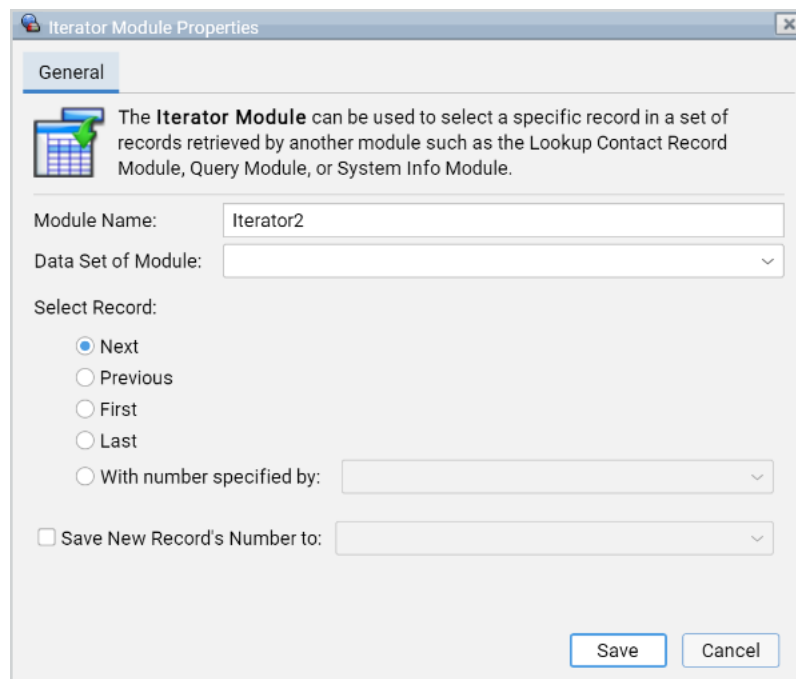
The screenshot shows the 'Input Module Properties' dialog box with the 'Text' tab selected. The dialog has tabs for General, Prompts, Events, Confirmation, Page, Text, and Disposition. In the Text tab, there is a checkbox for 'Use Visual IVR settings' which is unchecked, and a sub-checkbox for 'Text only' which is checked. Below these is a large table with two columns: 'Type' and 'Text / File Name'. The table is currently empty. At the bottom of the dialog, there are buttons for 'Add' (with a dropdown arrow), 'Remove', 'Edit', 'Up', 'Down', 'Convert to ML Prompt', 'Save', and 'Cancel'. A small menu is open under the 'Add' button, showing 'Add Text' and 'Add Multilingual Prompt'.

Dispositions tab. The default disposition is Caller Disconnected. For more information, see [Selecting Dispositions](#).

The screenshot shows the 'Input Module Properties' dialog box with the 'Disposition' tab selected. The dialog has tabs for General, Prompts, Events, Confirmation, Page, Text, and Disposition. In the Disposition tab, there is a label 'Disposition for this flow:' followed by a list box containing 'Caller Disconnected'. At the bottom of the dialog, there are buttons for 'Add', 'Remove', 'Save', and 'Cancel'.

Iterator Module

The Iterator module enables you to select a specific contact record from a set of records received from other modules.



The screenshot shows the 'Iterator Module Properties' dialog box with the 'General' tab selected. The dialog contains a description of the module's function, input fields for 'Module Name' (set to 'Iterator2') and 'Data Set of Module', a 'Select Record' section with radio buttons for 'Next' (selected), 'Previous', 'First', 'Last', and 'With number specified by', and a checkbox for 'Save New Record's Number to'. 'Save' and 'Cancel' buttons are at the bottom right.

- **Data Set of Module:** List of records retrieved by another module, such as the Lookup Contact Record, Query, or System Info modules.
- **Select record:**
 - Next
 - Previous
 - First
 - Last
 - With number specified by: List of call-attached variables
- **Save New Record's Number to:** Whether to save a record in another call variable. Because you may not use call variables that contain sensitive information in this field, these variables do not appear in the menu even if they exist in the Call Variables folder of the VCC.

IVA Transfer Module

The Intelligent Virtual Agent (IVA) transfer module enables you to pass interaction control to a chatbot, such as Five9 IVA-Digital, with an option to escalate that interaction to an agent when necessary. This section describes how to configure the IVA transfer module and provides an example IVR script that you can download and use in your environment.

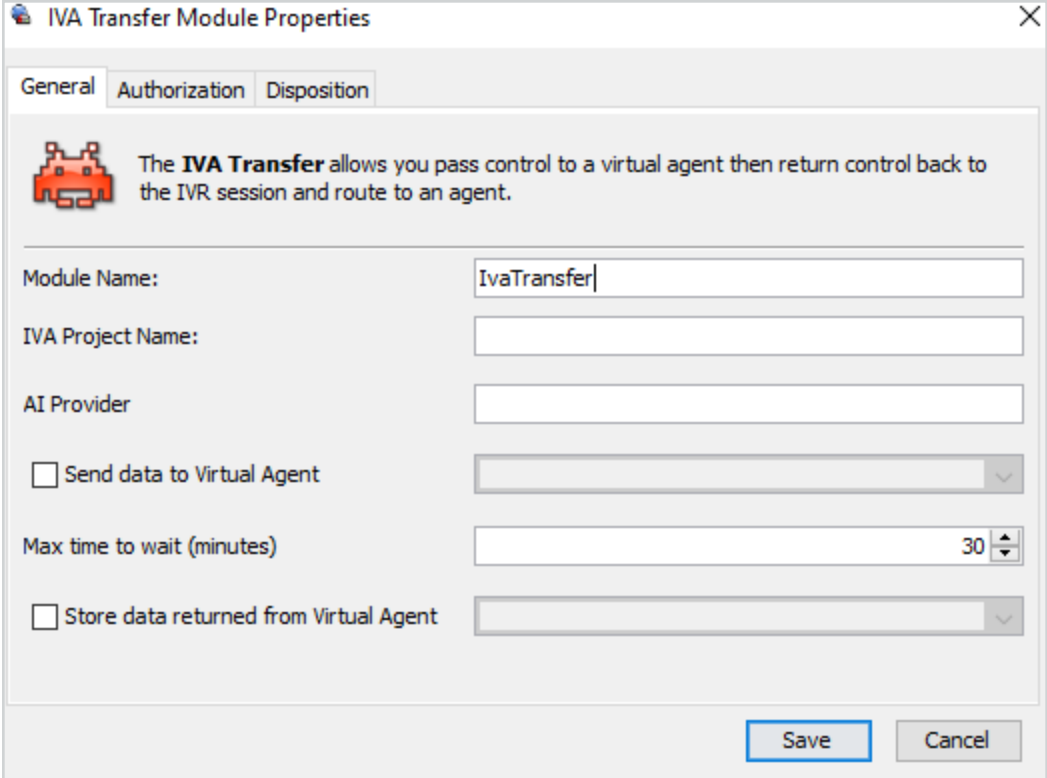
A pre-configured basic authorization profile is required to populate the [Authorization Tab](#) below. For more information about [creating authorization profiles](#), see the [Basic Configuration Administrator's Guide](#).

To control your IVA hours of operation, see the chat profile business hours settings in the [Digital Engagement Administrator's Guide](#).

Properties

[General tab](#)
[Authorization Tab](#)
[Disposition Tab](#)

General tab



The IVA Transfer Module Properties dialog box is shown with the General tab selected. It features a red robot icon and a description: "The **IVA Transfer** allows you pass control to a virtual agent then return control back to the IVR session and route to an agent." Below this, there are several fields: "Module Name" (containing "IvaTransfer"), "IVA Project Name", "AI Provider", a checkbox for "Send data to Virtual Agent", "Max time to wait (minutes)" (set to 30), and another checkbox for "Store data returned from Virtual Agent". At the bottom right are "Save" and "Cancel" buttons.

Field	Value
Module Name:	IvaTransfer
IVA Project Name:	
AI Provider	
☐ Send data to Virtual Agent	
Max time to wait (minutes)	30
☐ Store data returned from Virtual Agent	

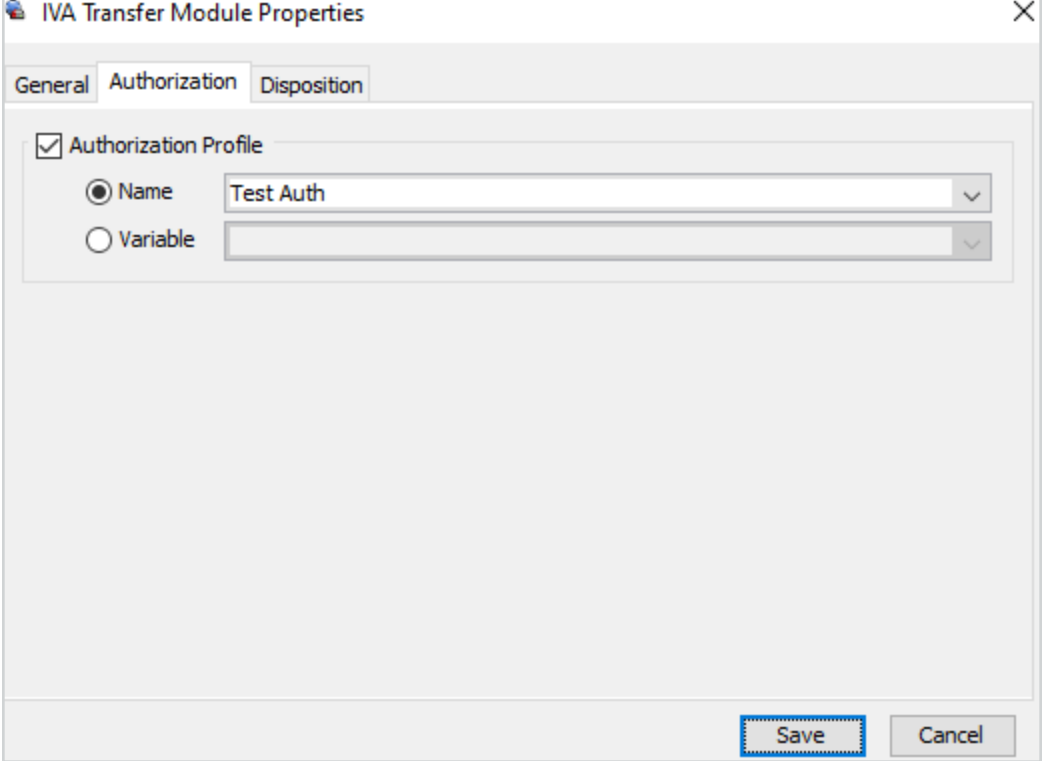
- **Module Name:** Name shown in IVR builder.
- **IVA Project Name:** Customer-facing name shown in your chatbot agent.
- **AI Provider:** URL to your chatbot vendor.

Example: <https://interaction.us7.studioportal.io/tasks/<your-chatbot-task-key-here>>

- **Send data to Virtual Agent:** Select check box to pass contact variables to the chatbot.
- **Max time to wait (minutes):** Maximum number of minutes allowed for this interaction. This setting controls the messaging bot time-out message that is configured in the Five9 Digital Engagement Administrator console.

- **Store data returned from Virtual Agent:** Select check box to store the variable information from the chatbot.

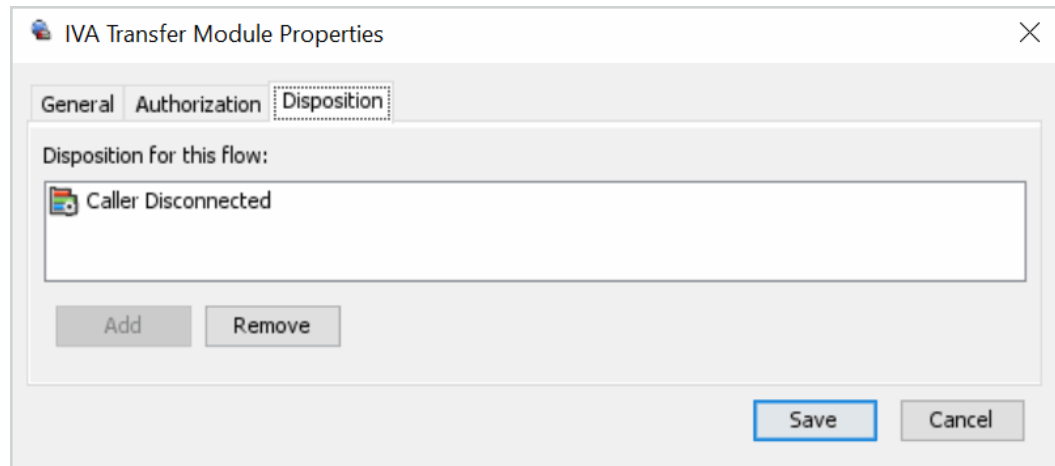
Authorization Tab



The screenshot shows the 'IVA Transfer Module Properties' dialog box with the 'Authorization' tab selected. The 'General' tab is also visible. The 'Authorization Profile' section is checked. Under this section, the 'Name' radio button is selected, and the text 'Test Auth' is entered in the adjacent text box. The 'Variable' radio button is unselected, and its text box is empty. At the bottom right, there are 'Save' and 'Cancel' buttons. The 'Save' button is highlighted with a blue dashed border.

- **Authorization Profile:**
 - Name: Name of an authorization profile configured for chatbot in VCC.
 - Variable: Name of a variable specified for chatbot authorization.
- **Possible Errors:** Chatbot is unavailable.

Disposition Tab



The screenshot shows a dialog box titled "IVA Transfer Module Properties" with a close button (X) in the top right corner. It has three tabs: "General", "Authorization", and "Disposition", with the "Disposition" tab selected. Inside the "Disposition" tab, there is a section labeled "Disposition for this flow:" followed by a list box containing one item: "Caller Disconnected" with a small icon to its left. Below the list box are two buttons: "Add" and "Remove". At the bottom right of the dialog box are two buttons: "Save" and "Cancel".

The default disposition is **Caller Disconnected**. Select a custom disposition for interaction flows that end in [Five9 Engagement Workflow](#). For more information, see [Selecting Dispositions](#).

Chat Customer Disconnect or Timeout

A chat user (customer/contact) disconnect or timeout may cause the IVR to fail at the next transfer, as the chat interaction is already disconnected. To enable post-terminate options, such as disposition, email transcript, survey, and so on, configure a hangup module as the next step after a disconnect. See `chatbot_example` script below.

Using the Chatbot Script

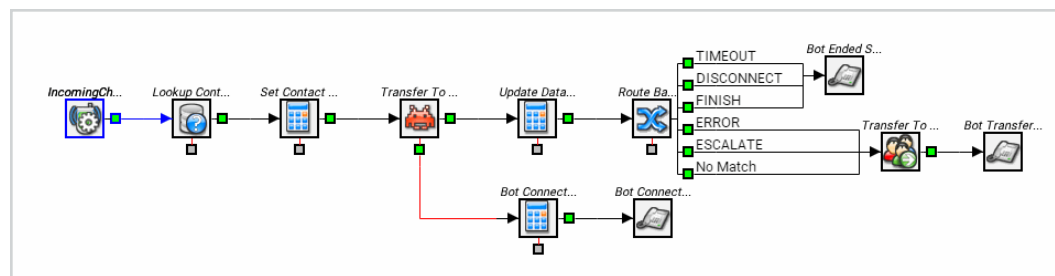
The supplied script assumes that you have a basic working knowledge of the Five9 IVR Builder. In this example, the contact information is retrieved from your system, if available, and sent to your chatbot. Any information returned from your chatbot is stored in Five9 as a chat transcript, and the interaction is routed by using standard Five9 IVR functionality.

You can rename the module and assign specific parameters in the **General** tab. You use the **Authorization** tab to assign a pre-defined basic authorization profile for transfer to Five9 IVA-Digital for chatbot tasks. For tasks not using Five9IVA-Digital, you may also

use an OAuth2 authorization profile or use a variable for authorization. For more information about authorization profiles, see the [Basic Configuration Administrator's Guide](#).

You can download the chatbot example script [here](#) to save locally. Using an HTML editor, save the file with the `.five9ivr` extension for import into your Five9 IVR Builder. Use the Five9 IVR Builder to create a new IVR script, then import the local chatbot example script and modify for your domain. For more information, see [Importing IVR Scripts](#).

To configure the IVA Transfer module in your local script, right-click or double-click to access the module properties. You can configure multiple languages for localized support with unique chatbot instances used for each language.

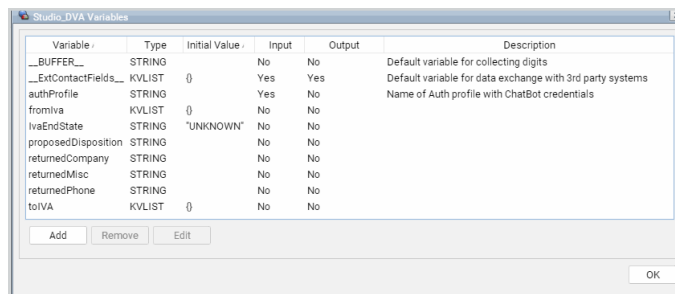


Creating the IVR Script Variables

The following variables are used in this IVR script:

Name	Type	Initial Value	Output	Input	Description
<code>_BUFFER_</code>	STRING		No	No	Default variable for collecting digits
<code>_ExtContactFields_</code>	KVLIST	<code>{ }</code>	Yes	Yes	Default variable for data exchange with a third party system
<code>authProfile</code>	STRING		No	Yes	Name of AuthProfile with chatbot

Name	Type	Initial Value	Output	Input	Description
fromIva	KVLIST	{ }	No	No	credentials Data return from chatbot
IvaEndState	STRING	"UNKNOWN"	No	No	Identifies chatbot completion
proposedDisposition	STRING		No	No	
returnedCompany	STRING		No	No	
returnedMisc	STRING		No	No	
returnedPhone	STRING		No	No	
toIva	KVLIST	()	No	No	Data sent to chatbot



For more information about creating and assigning your IVR script variables, see [Managing Variables](#).

Looking up Contact Information

This script begins by using the Lookup Contact Record Module to identify your inbound contact information using first name, last name, and email fields. Unidentified contacts may be sent to a live agent later in this script. You can customize your contact record lookups to suit your needs. Phone numbers provided to the chatbot may be added to your callback list.

The **Lookup Contact Record Module** can be used to identify Contact Records that match specific criteria, such as the caller's phone number or an account number provided by the caller.

Module Name:

☒ Lookup Record Based on Contact Fields Match:

☐ Lookup Record Based on Condition:

☐ Use Call Attached Contact Record

If Multiple Records Match the Condition:
☒ Raise an Exception
☐ Save Number of Records to:

Fetch timeout (sec):

Storing and Using Your Contact Information

Use the Set Variable Module to store the variable information from the previous step in a KVList to send to the chatbot.

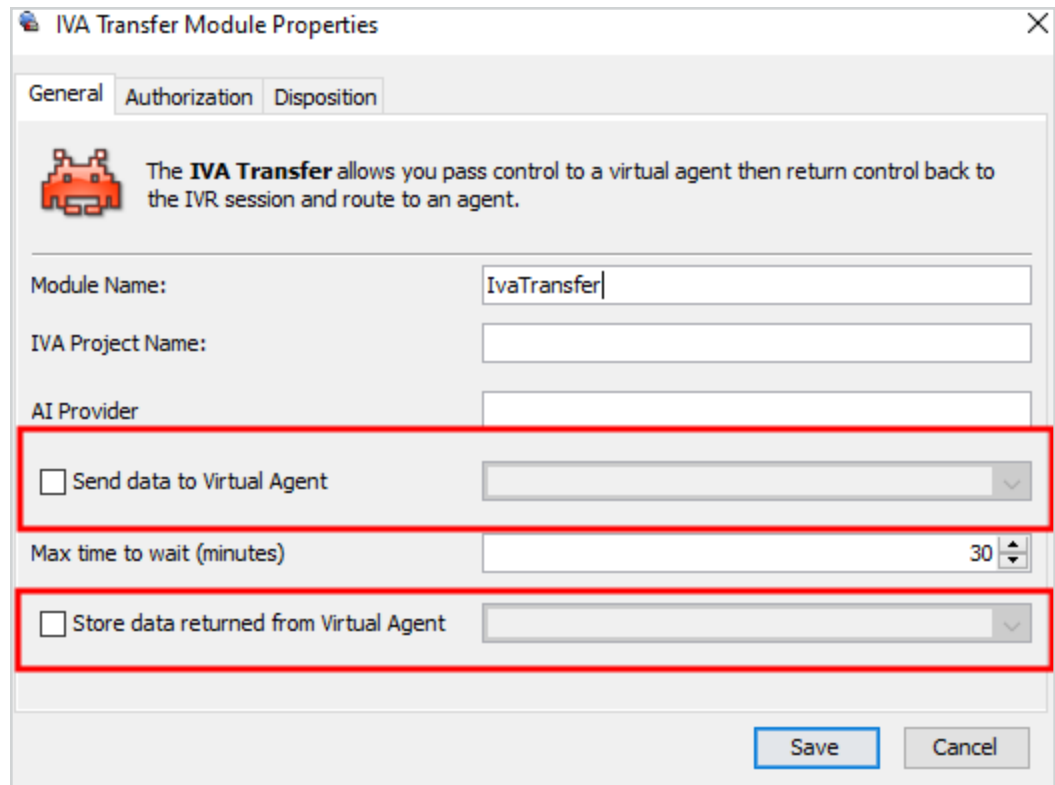
The **Set Variable Module** can be used to set or update the value of an IVR Variable or Call Variable.

Module Name:

Variable	Assigned Variable, Constant, or Function
__BUFFER__	PUT(toIVA, "contact_name", Omni.name)
__BUFFER__	PUT(toIVA, "contact_number", Contact.number1)
__BUFFER__	PUT(toIVA, "contact_company", Contact.company)

Sending the Interaction to the Chatbot

Use the IVA Transfer module to send the interaction and any variable information that you have identified to the chatbot using the **to/fromIVA** script variables configured in the first section.



The IVA Transfer Module Properties dialog box is shown with the 'General' tab selected. It contains a description of the IVA Transfer module, which allows passing control to a virtual agent and returning control to the IVR session. The dialog includes fields for 'Module Name' (IvaTransfer), 'IVA Project Name', and 'AI Provider'. Two options are highlighted with red boxes: 'Send data to Virtual Agent' and 'Store data returned from Virtual Agent', both of which are currently unchecked. A 'Max time to wait (minutes)' field is set to 30. 'Save' and 'Cancel' buttons are at the bottom right.

IVA Transfer Module Properties

General Authorization Disposition

 The **IVA Transfer** allows you pass control to a virtual agent then return control back to the IVR session and route to an agent.

Module Name: IvaTransfer

IVA Project Name:

AI Provider:

☐ Send data to Virtual Agent

Max time to wait (minutes): 30

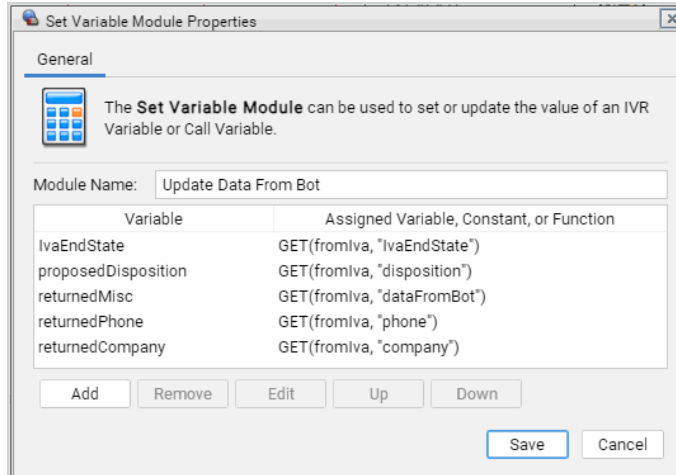
☐ Store data returned from Virtual Agent

Save Cancel

Following the chatbot involvement, you can use logic within the IVR script to set the appropriate disposition, such as within the existing case statement, hangups actions, and so on, depending on your final configuration.

Storing the Returned Variable Information

Use the second Set Variable Module to store any information returned from the chatbot to the script variables defined previously.



The **Set Variable Module** can be used to set or update the value of an IVR Variable or Call Variable.

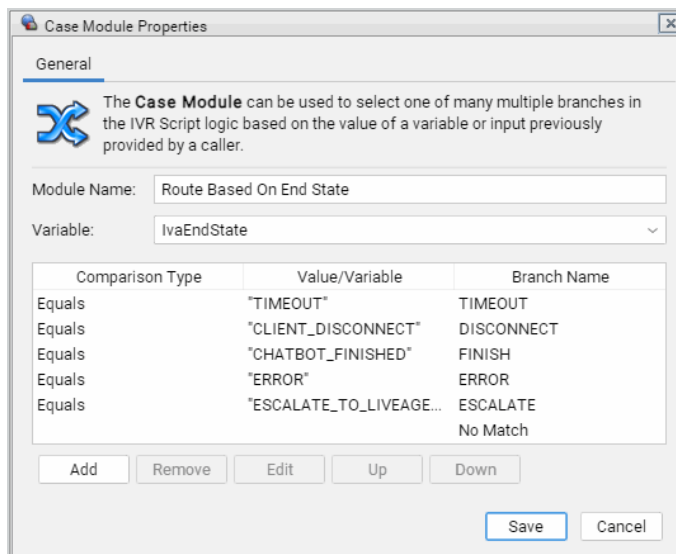
Module Name: Update Data From Bot

Variable	Assigned Variable, Constant, or Function
IvaEndState	GET(fromIva, "IvaEndState")
proposedDisposition	GET(fromIva, "disposition")
returnedMisc	GET(fromIva, "dataFromBot")
returnedPhone	GET(fromIva, "phone")
returnedCompany	GET(fromIva, "company")

Buttons: Add, Remove, Edit, Up, Down, Save, Cancel

Continuing the Interaction

Use the Case Module to determine any next steps for this interaction. This module enables the full flexibility of traditional interaction routing to assist your customer as needed.



The **Case Module** can be used to select one of many multiple branches in the IVR Script logic based on the value of a variable or input previously provided by a caller.

Module Name: Route Based On End State

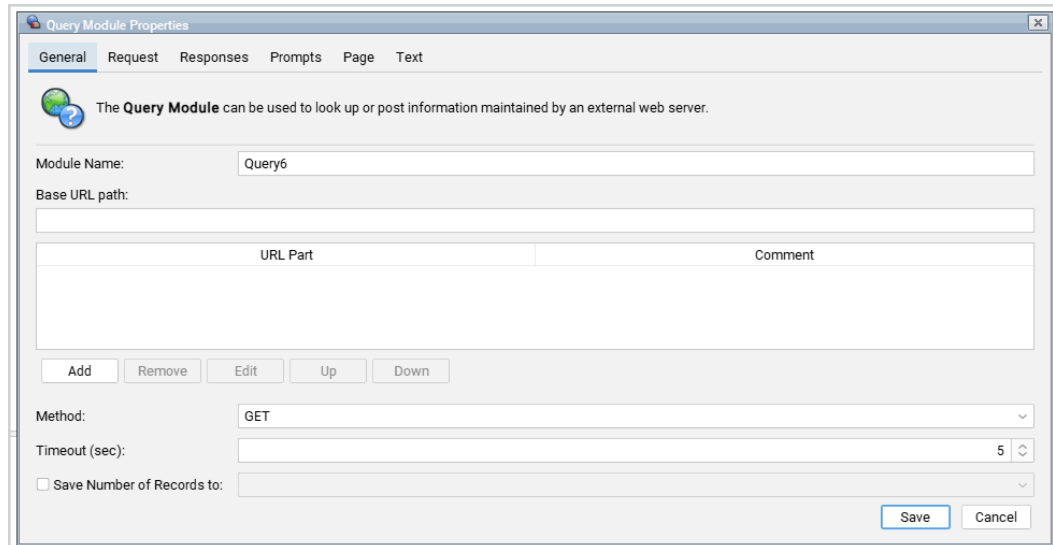
Variable: IvaEndState

Comparison Type	Value/Variable	Branch Name
Equals	"TIMEOUT"	TIMEOUT
Equals	"CLIENT_DISCONNECT"	DISCONNECT
Equals	"CHATBOT_FINISHED"	FINISH
Equals	"ERROR"	ERROR
Equals	"ESCALATE_TO_LIVEAGE..."	ESCALATE
		No Match

Buttons: Add, Remove, Edit, Up, Down, Save, Cancel

Adding New Contact Information

In this example, a Query Module is used to add new contact information to a dialing list.



The **Query Module** can be used to look up or post information maintained by an external web server.

Module Name:

Base URL path:

URL Part	Comment
----------	---------

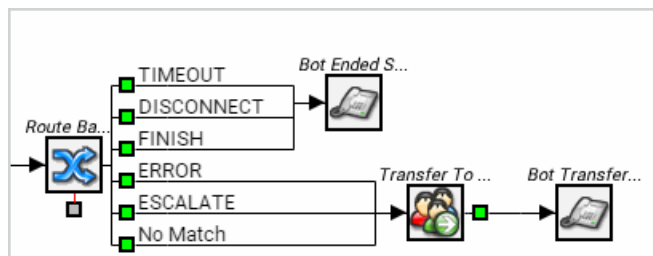
Method:

Timeout (sec):

☐ Save Number of Records to:

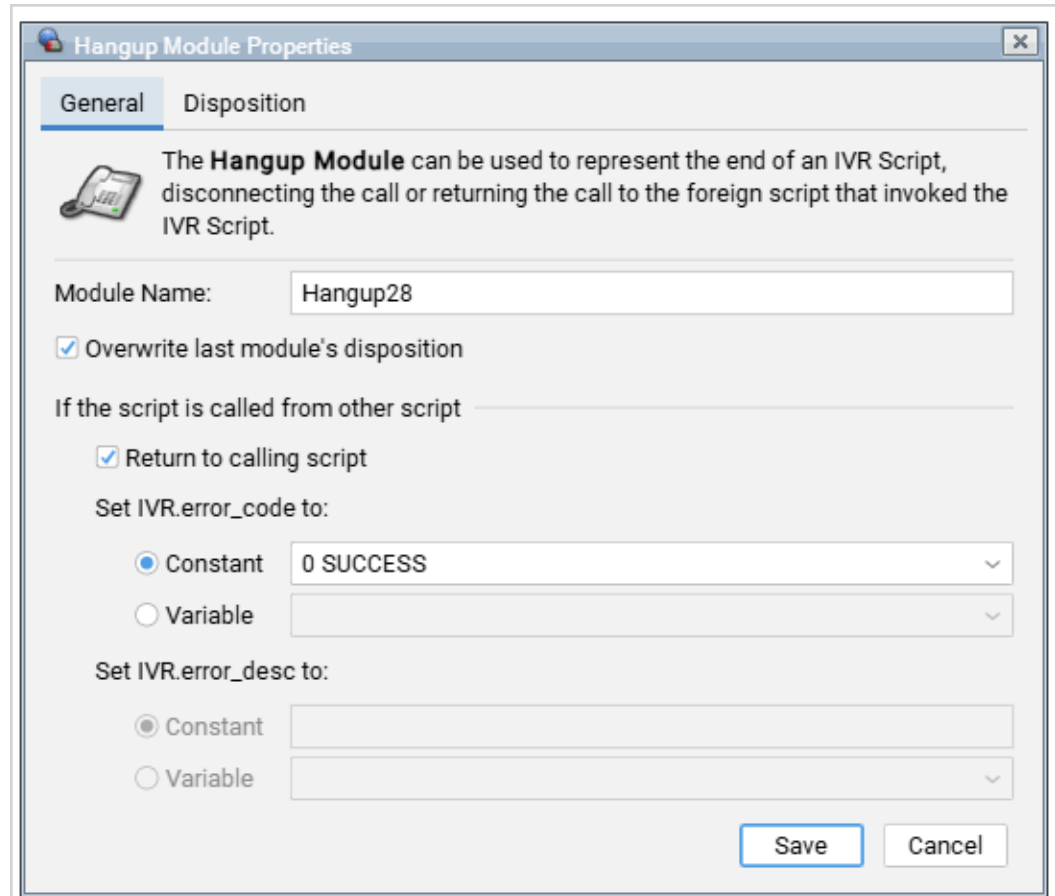
Sending the Interaction to an Agent

You can continue to route the interaction or assign a disposition to end the interaction. This example shows the Skill Transfer Module used to further route this interaction.



Ending the Interaction

You can use a hangup module to assign a disposition and end the interaction. In the General tab, check **Overwrite last module's disposition** to activate the **Disposition** tab.



The **Hangup Module** can be used to represent the end of an IVR Script, disconnecting the call or returning the call to the foreign script that invoked the IVR Script.

Module Name:

☒ Overwrite last module's disposition

If the script is called from other script

☒ Return to calling script

Set IVR.error_code to:

☒ Constant

☐ Variable

Set IVR.error_desc to:

☒ Constant

☐ Variable

Language Module

Use this module to assign language routing options based on the languages defined in your Script menu (Call.Language variables) for use in IVR scripts. See [Languages](#) for how to add Call.Language variables to the Script menu.

[General Tab](#)
[Prompts Tab](#)
[Events Tab](#)
[Dispositions Tab](#)

General Tab

The screenshot shows the 'Language Module Properties' dialog box with the 'General' tab selected. The dialog has four tabs: General, Prompts, Events, and Disposition. A green checkmark icon is next to the text: 'The **Language** module can be used to set a value to Call.language variable, so that prompts specific to the variable will be played in the following module.'

Module Name:

☐ Set to:

- ☒ Constant
- ☐ Variable

☒ On Choice:

Max Attempts:

Choice	DTMF	Visual Mode	Thumbnail
en-US	1	Yes	

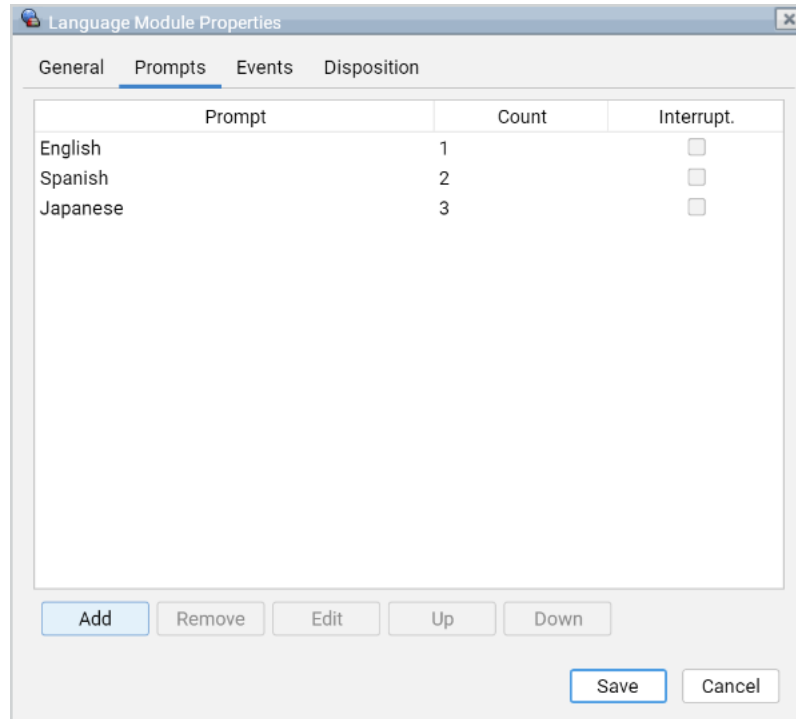
Buttons: Add, Remove, Edit, Save, Cancel

- **Module Name:** Role of the module in the IVR script. In Visual IVR scripts, the name is converted to text that appears in the heading of the IVR window.

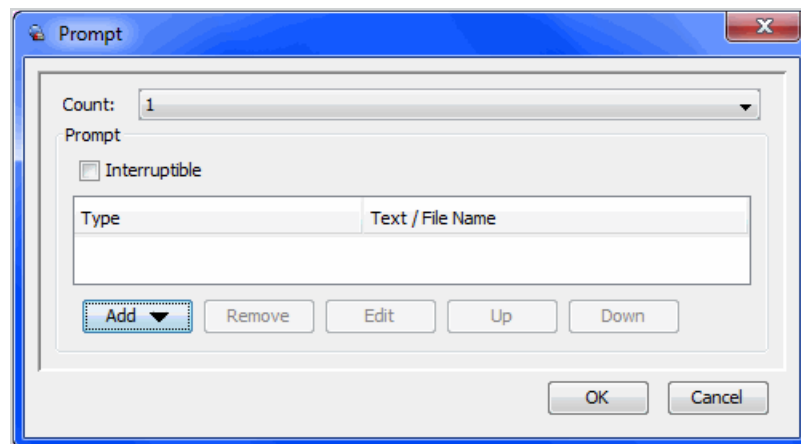
- **Set To:** Whether to assign a constant or variable language option.
 - Constant (default setting): A single language is offered. The module functions as a simple assignment. This option disables the Prompts and Events tabs. No prompts are played and no inputs are expected in this case.
 - Variable: Language options determined by your Call.language variable options.
- **On Choice:** Language items from which callers can select:
 - Choice: Language variable options correspond to languages defined in the Script menu. At least one choice must be created when using this option.
 - DTMF: Phone key that corresponds to the language selection. You can use a specific digit or automatic assignment.
 - Visual Mode: Whether the branch is enabled in the Visual IVR script.
- **Max Attempts:** Maximum number of unsuccessful attempts at recognizing the language selection, after which a module exception exit is activated. When a NoInput or NoMatch error occurs, the prompt is repeated for up to the selected number of times. If this number is exceeded, the module is exited with an error code. This option enables you to limit the number of attempts.

Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). Each prompt can be interruptible.



In Visual IVR scripts, TTS prompts are converted to text that appears in the body of the IVR window.



- **Count:** Minimum number of times that the user must speak for the prompt to be played. The default is 1.

- **Interruptible:** Whether to stop playing the prompt as soon as the caller has completed the requested action.
 - Enabled: Stop playing the prompt as soon as the caller has completed the action.
 - Disabled: Continue and finish playing the prompt even if the caller has completed the action.

Events Tab

List of event handling options. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in a specific sequence.

The screenshot shows the 'Language Module Properties' dialog box with the 'Events' tab selected. The dialog has four tabs: General, Prompts, Events, and Disposition. The 'Events' tab contains a table with the following data:

Event	Count	Event's Prompt	Interrupt.	Action
NoMatch	1	NoMatchPrompt	☐	continue
NoInput	1	NoInputPrompt	☐	reprompt

Below the table are three buttons: 'Add', 'Remove', and 'Edit'. At the bottom of the dialog, there are three input fields with spinners:

- Max Time to Enter (secs): 5
- No Input Timeout (secs): 5
- Recognition Confidence Threshold (%): 0

At the bottom right are 'Save' and 'Cancel' buttons.

- **Event:** List of possible events:

The screenshot shows a window titled "Event" with the following fields and controls:

- Event:** A dropdown menu showing "NoInput".
- Count:** A dropdown menu showing "1".
- Action:** A dropdown menu showing "continue".
- Prompt:** A section containing:
 - An unchecked checkbox labeled "Interruptible".
 - A table with two columns: "Type" and "Text / File Name". The table is currently empty.
 - Buttons: "Add", "Remove", "Edit", "Up", "Down", and "Convert to ML Prompt".
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

- **NoMatch:** Generated if the speech recognition engine cannot find a match in the grammar.
 - **NoInput:** Generated if the speech recognition engine does not hear any caller input before the time-out.
 - **Help:** Generated if the caller presses the Help button.
- **Count:** Number of possible instances (1–13) for an action, which enables you to handle differently multiple errors of the same type. The count is increased regardless of the event.

Example

NoMatch event: The first time, the module repeats the prompt to the caller. The second time, the IVR exits the module.

- **Match:** Whether the match should be exact or approximate.
 - **Appr:** Additional words after the grammar item are allowed if the required words are recognized, for example: The prompt *[For apple juice, say Apple or press 1. For orange juice, say Orange or press 2.]* enables

callers to say *Apple, please* or *Apples of course* instead of only *Apple*.

- **Exact:** Grammar items must be recognized completely.
- **DTMF:** Phone key that corresponds to the grammar phrase. You can use a specific digit or automatic assignment. With automatic assignment, each option corresponds to the previous option plus 1. For example, with the prompt [*For apple juice, say Apple or press 1. For orange juice, say Orange or press 2.*], 1 and 2 are assigned automatically if you select <Auto>.
- **Thumbnail URL:** Optional URL of a thumbnail image to appear in that branch.
 - **Value:** Specific value.
 - **Variable:** Name of variable. The value is the URL of the image.
- **On Recognition/DTMF:** Branch to use for each successful or failed event.
 - **Leave Module:** Content of the Branches tab and the No Match branch, which is used when the comparison does not match one of the options, such as when the prompt specifies 1 or 2, but the caller presses 3 or nothing.
 - **Through Event:** List of events described in the **Events** tab.
- **Count:** Minimum number of times that the user must speak for the prompt to be played. The default is 1.
- **Max Time to Enter:** Number of seconds that the module waits for callers to finish entering the information. If callers exceed the time, the IVR script proceeds to the next module or repeats the current module. The default is 15 seconds.

Dispositions Tab

The default disposition is Caller Disconnected. For more information, see [Selecting Dispositions](#).

Lookup Contact Record Module

The Lookup Contact Record module enables you to retrieve information from the caller's contact record. This information can be used to process the call and to associate the call with the record by using the Contact Update module. The module

branches are success (at least one contact record exists) or failure (no contact record exists).

When using Visual IVR scripts, you may also add images to pages and menus to enhance your brand or to describe a product or service. However, you cannot add links to images. For more information, see [Page Tab](#).

[General Tab](#)

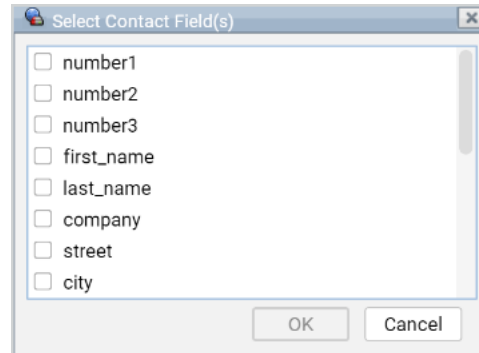
[Prompts Tab](#)

[Page Tab](#)

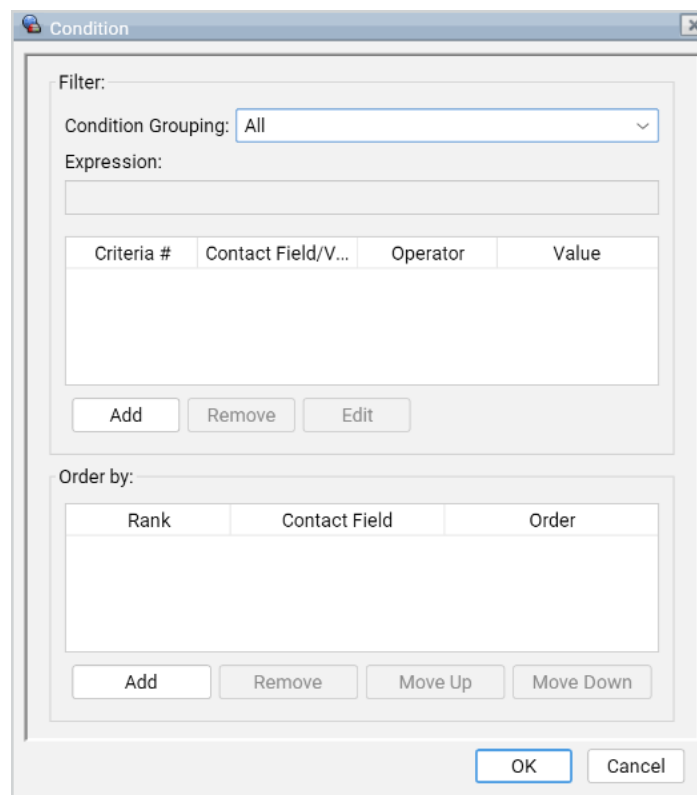
The screenshot shows the 'Lookup Contact Record Module Properties' dialog box with the 'General' tab selected. The dialog has a title bar and four tabs: General, Prompts, Page, and Text. Below the tabs is a description: 'The Lookup Contact Record Module can be used to identify Contact Records that match specific criteria, such as the caller's phone number or an account number provided by the caller.' Below this is a 'Module Name' field containing 'LookupContactRecord3'. There are three radio button options: 'Lookup Record Based on Contact Fields Match:' (selected), 'Lookup Record Based on Condition:', and 'Use Call Attached Contact Record'. Each has a corresponding text field and a dropdown arrow. Below these is a section 'If Multiple Records Match the Condition:' with two checkboxes: 'Raise an Exception' (checked) and 'Save Number of Records to:' (unchecked). At the bottom is a 'Fetch timeout (sec):' field with a value of 60 and a dropdown arrow. 'Save' and 'Cancel' buttons are at the bottom right.

General Tab

- **Module Name:** Name that represents the role of the module in the IVR script. In Visual IVR scripts, the name is converted to text that appears in the heading of the IVR window.
- **Lookup Record Based on Contact Fields Match:** Contact field to use for the search. The menu contains a list of your contact fields.

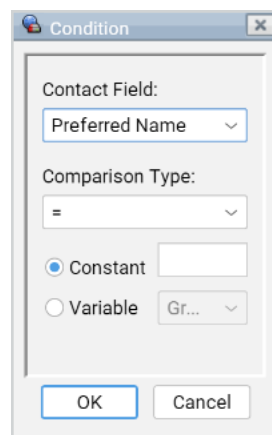


- **Lookup Record Based on Condition:** Contact field to use for the search. The menu contains a list of variables to create conditions.



- **Condition Grouping:** Method used to join conditions:
 - Any: Any defined criteria are valid. The expression 1 OR 2 is entered automatically in the Expression field.

- All: All defined criteria must be valid. The expression 1 AND 2 is entered automatically in the Expression field.
 - Custom: Custom expression that describes how to use the criteria. This option enables the Expression field.
- **Expression:** Logical expression that describes how to compare the conditions.
- **Condition window:**



- Contact field: In the expression, left variable to be compared with a constant or another variable. List of your contact fields. Names are limited to 10 characters.
 - Comparison Type: List of possible methods for comparing the variables.
 - Constant: In the expression, optional specific value to compare to the contact field.
 - Variable: In the expression, optional right variable to compare to the contact field. List of your user, contact, system, agent, call, IVR, or user CAV variables.
- **Order by:** Up to three contact fields, by default starting with the first one alphabetically.
 - Contact Field: List of your contact fields. You can change the default to any other contact field.
 - Order: Ascending or descending
- **Use Call Attached Contact Record:** Record associated with the call is used instead of searching among all contact records. For Visual IVR, if multiple contact fields exist, all are concatenated with AND.

- **Raise an Exception:** If several records match the search criteria, whether to activate the exception exit.
- **Save Number of Records to:** If several records match the search criteria, whether to save the number of matched records in the selected integer CAV variable.
- **Fetch Timeout:** Maximum number of seconds for receiving a response from the script. After the time-out, the scripts exits the module.

Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). Each prompt can be interruptible.

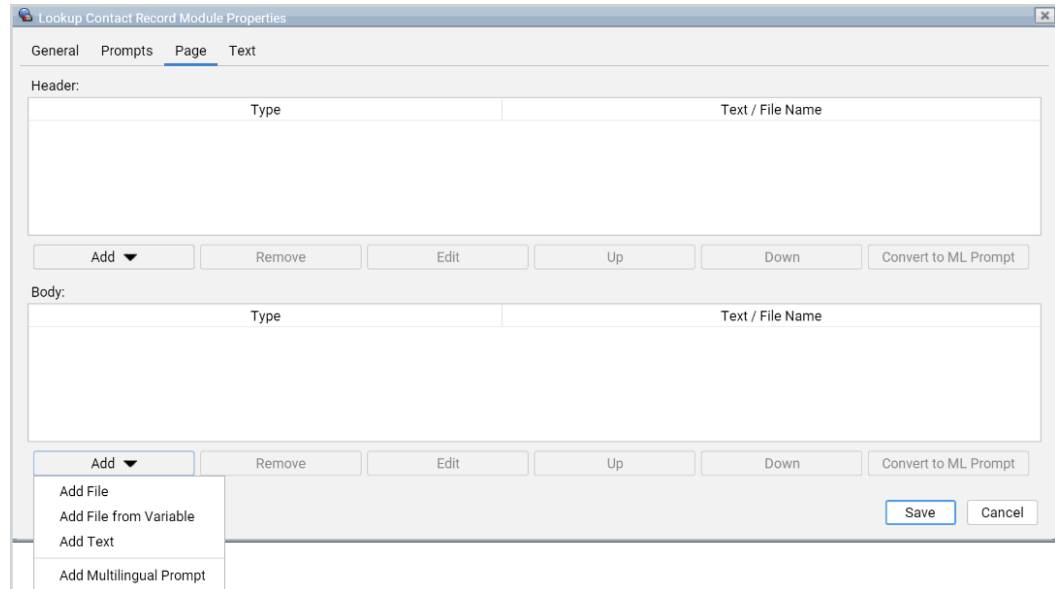
In Visual IVR scripts, TTS prompts are converted to text that appears in the body of the IVR window.

Page Tab

This tab contains the list of Visual IVR prompts that are shown visually in the customer's Web browser when a IVR script is executed.

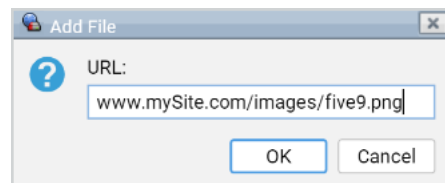
If this tab contains prompts, these are shown whereas prompts in the Prompts tab are not.

If this tab does not contain prompts, the prompts in the Prompts tab are shown.

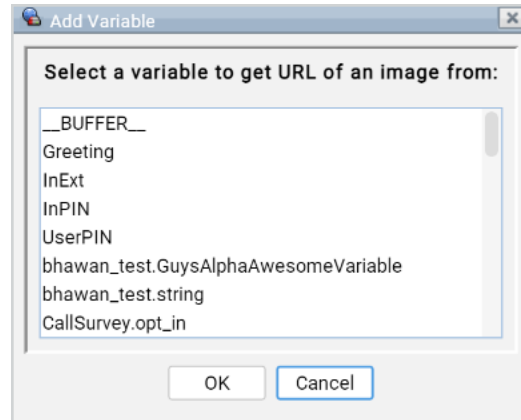


This tab enables you to add different types of prompts:

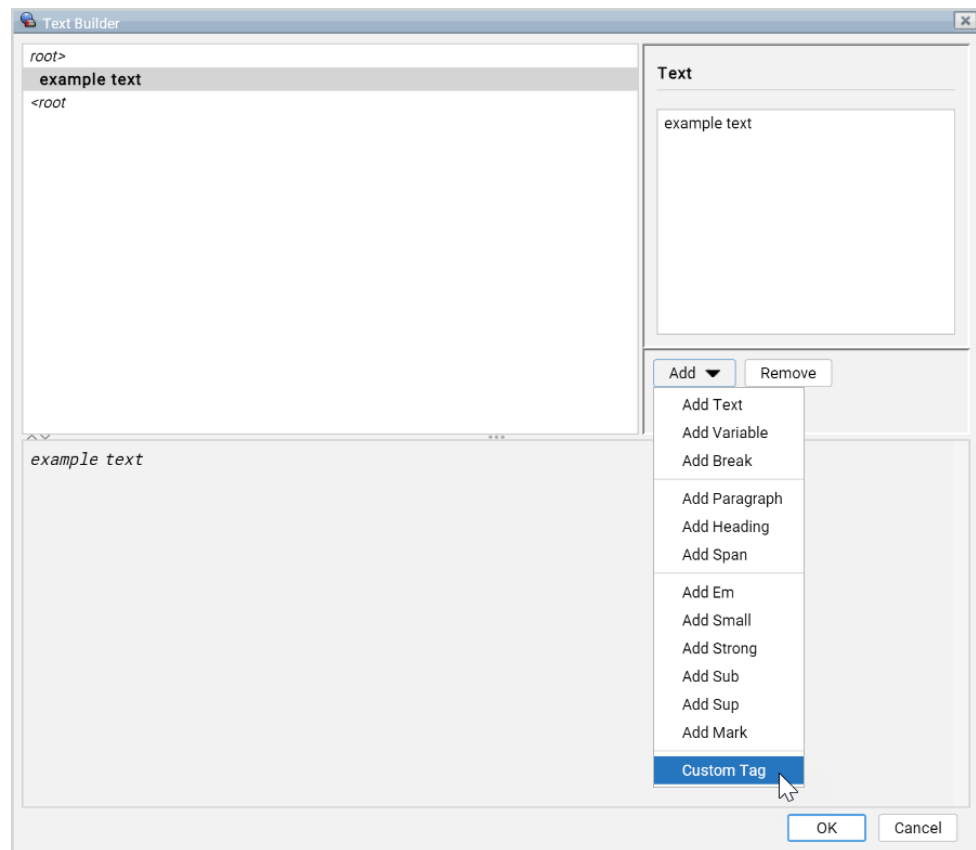
- **File:** URL of an image.



- **Text:** Formatted text that can contain variables. You cannot add images. If you select a variable that contains the URL of an image, the URL is displayed.
- **Variable:** Variable that contains the URL of an image.



- **Custom Tag:** See detailed usage information in [Using the Visual IVR URL](#).



The text editor enables you to create HTML text that contains formatted text and variables. You can customize the text with bold, color, and emphasis. Unlike breaks in

the TTS builder, which are associated with speech, breaks in this text editor add a line or horizontal break. The Test button enables you to preview the HTML text.

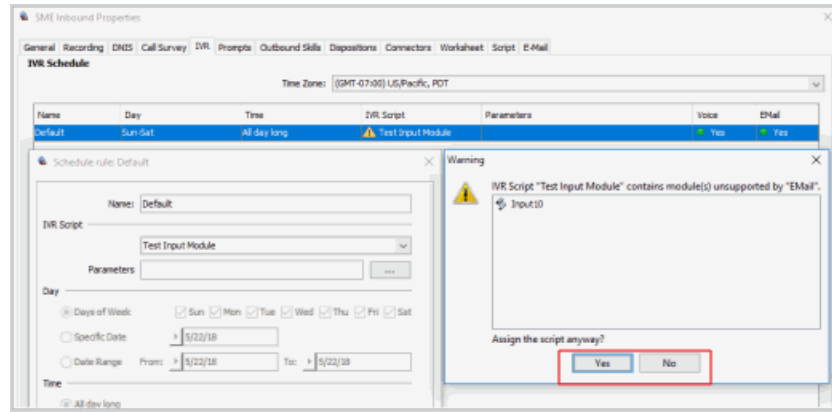


Menu Module

This module enables you to configure multiple exit branches (one branch for each case) depending on the caller's selection. The text spoken by the caller is analyzed with grammar different from the grammar used by the Input Module. The result is compared to phrases expected by the module.

Digital Channels

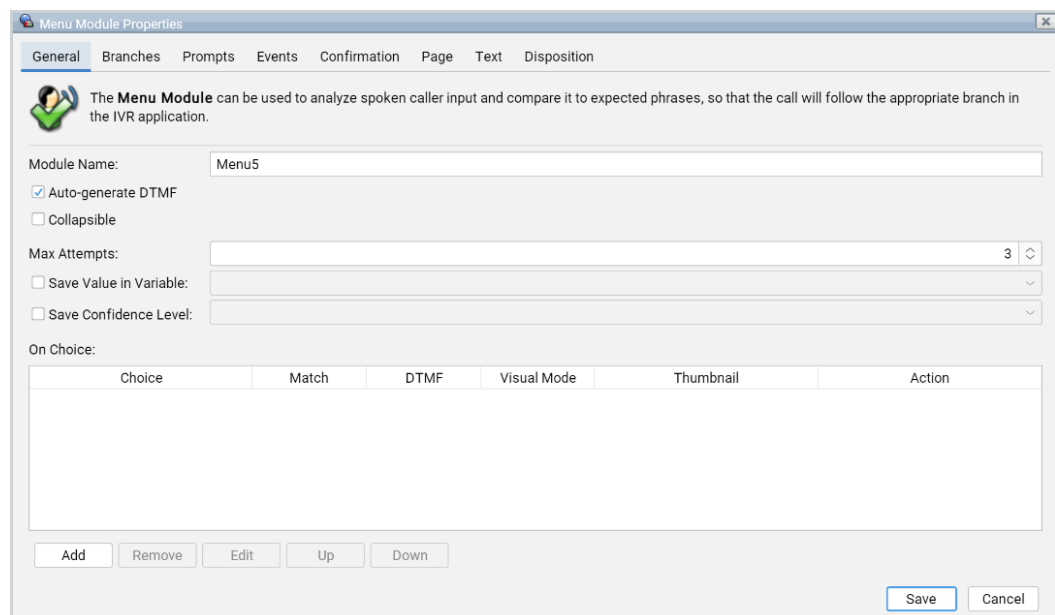
This module is not compatible with digital channels. If your domain is enabled for digital channels, you see a warning when you use IVR scripts that contain this module in inbound campaigns. If you choose to use the script, a warning icon is displayed in the schedule description. The IVR schedule still works, but these incompatible modules are skipped.



Properties

The Menu module is equivalent to the Get Digits module followed by the Case module.

[General Tab](#)
[Branches Tab](#)
[Prompts Tab](#)
[Events Tab](#)
[Confirmation Tab](#)
[Page Tab](#)
[Dispositions Tab](#)



General Tab

- **Module Name:** Name that represents the role of the module in the IVR script. In Visual IVR scripts, the name is converted to text that appears in the heading of the IVR window.
- **Use Speech Recognition:** Whether to enable the speech recognition.

Note: If you need speech recognition features, contact your Five9 account director.

- **Auto-Generate DTMF:** Whether to use of phone keys to choose menu items.
The Menu module supports DTMF and speech input. However, in some cases, you should use only one type of input.
 - Enabled: Use user-defined key assignment, or delegate the assignment (0-9, #, or *) to the module (<Auto>).
 - Disabled: No DTMF is accepted; speech only.
- **Record User's Input as Audio File:** Whether to save the voice input as audio file to access from reports. This is useful for evaluating how well your grammar matches callers' spoken phrases. The recording is available in reports but is not sent to agents or skill groups. See also [Using Recorded Files](#).
- **Collapsible:** Visual IVR feature that applies to all types of prompts. Whether to concatenate all the prompts in the module whether they are played (standard prompts) or displayed (Visual IVR prompts) to callers.
 - Enabled: Concatenate prompts. Callers see at once all the prompts in the same window.
 - Disabled: Do not concatenate prompts. Callers hear or see each prompt after performing the requested action.
- **Max Attempts:** Maximum number of unsuccessful attempts at recognizing the speech, after which an module exit exception activates. When a NoInput or NoMatch error occurs, the prompt repeats for up to the set number of times. Exceeding this number exits the module using the No Match branch. This limits the number of recording attempts. You must link the error branch to subsequent IVR modules to handle the error.
- **Save Value in Variable:** Whether to save the recognized input in an IVR or CAV string variable.

- **Save Confidence Level:** Whether to save the returned confidence level.

A confidence value variable is associated with every speech recognition attempt. A successful recognition indicates that your confidence threshold has been met. An unsuccessful recognition also returns the highest scoring choice and the associated confidence value in separate variables. Both variables are available in Five9 reports. For more information, see the [Dashboards and Reports User's Guide](#).

- **On Choice:** Required option: At least one Leave Module option

List of grammar items from which callers can select.

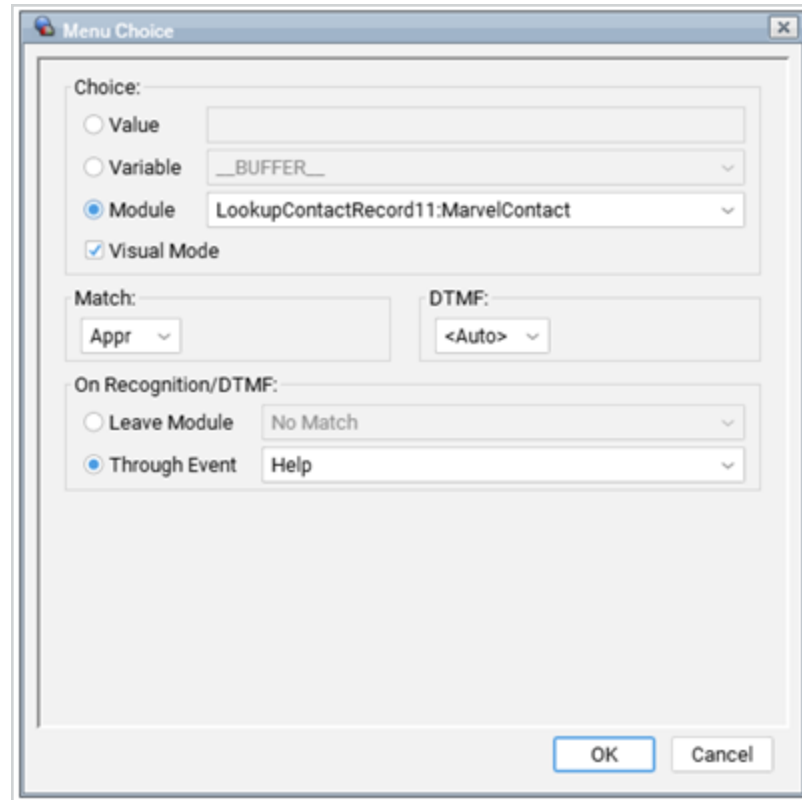
- **Choice:** Grammar phrases that activate the actions. The grammar can be text, variables, or the results of the Query and Contact Lookup modules specified by a module name and field. The phrases can be used in generated prompts to create speech recognition grammar.
- **Match:** Whether the match to the grammar can be approximate or must be exact.
- **DTMF:** Phone key that corresponds to the grammar phrase. You can use a keypad options or automatic assignment.
- **Visual Mode:** Whether the branch is enabled in the Visual IVR script.
- **Thumbnail:** If Visual Mode is enabled, URL of the thumbnail image.
- **Action:** Module exit that is activated when the speech is recognized. You can specify only one exit for each item. If you need additional branches, use a Case module after the Menu module.

To add or modify a grammar item, click **Add** or **Edit**.

- **Menu Choice:** Menu that enables you to create or modify the grammar items that appear in the table of the General tab.

Important

Before using this menu, create the branches (see the [Branches Tab](#)) that you need to exit the module.

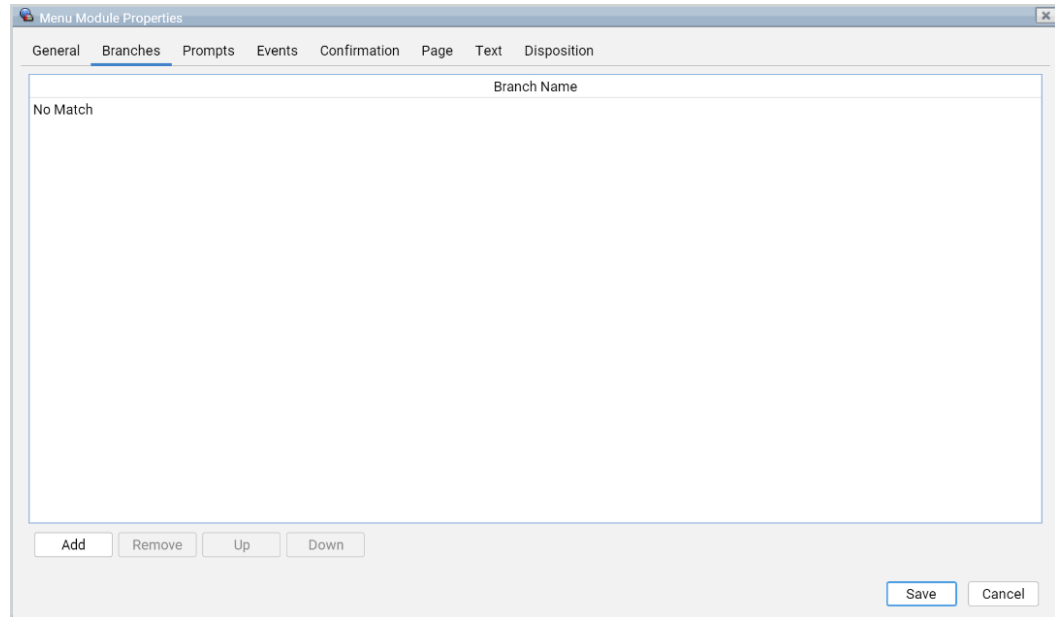


- **Choice:** Grammar item.
 - Value: Possible values, such as apples, cherries, and plums. Use this option as follows:
 - DTMF: Keep the caller's choice for further branching.
 - Speech recognition: Verify the caller's words, for example: *You have chosen {apples}. Is this correct?*
 - Variable: Name of IVR or CAV variable. The value is obtained from the variable.
 - Module: This option displays only when an IVR script contains Contact Lookup or Query modules to refer back to the results of the Contact Lookup or Query modules.
 - Visual Mode: Enable this feature so that your customer sees the image that you added to the script. However, in some cases, you may want to not show an image, especially if it is not relevant, such as when your customers want to use a voice IVR version of the script to reach your Support organization or when you create a master script that you customize for voice and Visual IVR campaigns.

- **Match:** Whether the match should be exact or approximate.
 - **Appr:** Additional words after the grammar item are allowed if the required words are recognized, for example: The prompt *[For apple juice, say Apple or press 1. For orange juice, say Orange or press 2.]* enables callers to say *Apple, please* or *Apples of course* instead of only *Apple*.
 - **Exact:** Grammar items must be recognized completely.
- **DTMF:** Phone key that corresponds to the grammar phrase. You can use any keypad option or automatic assignment. With automatic assignment, each option corresponds to the previous option plus 1. For example, with the prompt *[For apple juice, say Apple or press 1. For orange juice, say Orange or press 2.]*, 1 and 2 are assigned automatically if you select <Auto>.
- **Thumbnail URL:** Optional URL of a thumbnail image to appear in that branch.
 - **Value:** Specific value.
 - **Variable:** Name of variable. The value is the URL of the image.
- **On Recognition/DTMF:** Branch to use for each successful or failed event.
 - **Leave Module:** Content of the Branches tab and the No Match branch, which is used when the comparison does not match one of the options, such as when the prompt specifies 1 or 2, but the caller presses 3 or nothing. If the number of attempts is exceeded, the exception NOT_ENTERED occurs, and the script proceeds to the next module.
 - **Through Event:** List of events described in the [Events Tab](#).

Branches Tab

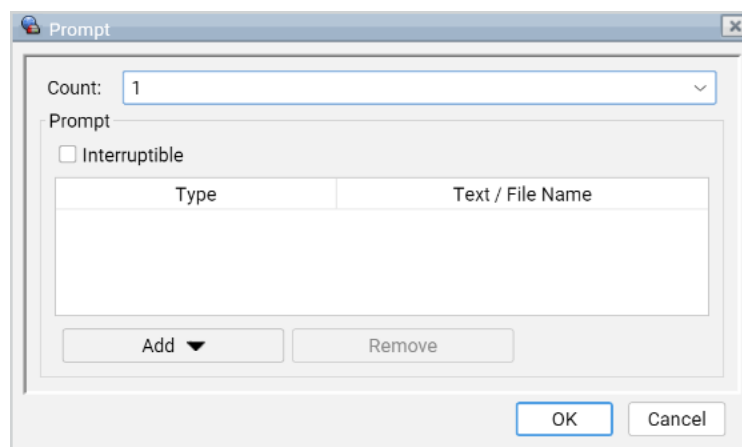
List of branches that exit the module. For example, when a caller speaks a phrase that contains the word *support*, the module exits the module through the Support branch. After you have created the exit branches, return to the General tab to create the grammar items.



Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). Each prompt can be interruptible.

In Visual IVR scripts, TTS prompts are converted to text that appears in the body of the IVR window.



- **Count:** Minimum number of times that the user must speak for the prompt to be played. The default is 1.
- **Interruptible:** Whether to stop playing the prompt as soon as the caller has completed the requested action.
 - Enabled: Stop playing the prompt as soon as the caller has completed the action.
 - Disabled: Continue and finish playing the prompt even if the caller has completed the action.

Events Tab

List of event handling options. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in a specific sequence.

The screenshot shows the 'Menu Module Properties' dialog box with the 'Events' tab selected. The dialog has tabs for General, Branches, Prompts, Events, Confirmation, Page, Text, and Disposition. The 'Events' tab contains a table with the following data:

Event	Count	Event's Prompt	Interrupt.	Action
NoMatch	1	NoMatchPrompt	☐	continue
NoInput	1	NoInputPrompt	☐	reprompt

Below the table are buttons for 'Add', 'Remove', and 'Edit'. At the bottom, there are two input fields: 'Max Time to Enter (secs):' with a value of 5 and a dropdown arrow, and 'No Input Timeout (secs):' with a value of 5 and a dropdown arrow. 'Save' and 'Cancel' buttons are at the bottom right.

- **Event:** List of possible events:

The screenshot shows a window titled "Event" with the following fields and controls:

- Event:** A dropdown menu with "NoInput" selected.
- Count:** A dropdown menu with "1" selected.
- Action:** A dropdown menu with "continue" selected.
- Prompt:** A section containing:
 - An ☐ labeled "Interruptible".
 - A table with two columns: "Type" and "Text / File Name". The table is currently empty.
- Buttons:** A set of buttons below the table: "Add" (with a dropdown arrow), "Remove", "Edit", "Up", "Down", and "Convert to ML Prompt".
- Footer:** "OK" and "Cancel" buttons.

- **NoMatch:** Generated if the speech recognition engine cannot find a match in the grammar.
 - **NoInput:** Generated if the speech recognition engine does not hear any caller input before the time-out.
 - **Help:** Generated if the caller presses the Help button.
- **Count:** Number of possible instances (1–13) for an action, which enables you to handle differently multiple errors of the same type. The count is increased regardless of the event.

Example: NoMatch event. The module repeats the prompt to the caller. If the number of attempts is exceeded, the exception NOT_ENTERED occurs, and the script proceeds using the No Match branch to the next module.

- **Action:** Action taken when the specified event occurs.

- Reprompt: Repeat the prompt listed in the Prompts tab.
 - Continue: Proceed to next attempt without playing the prompt listed in Prompts tab.
 - Exit: Exit by the NoMatch branch at the bottom of the module. This event can occur even if the maximum number of attempts has not been reached. The NoMatch branch must be linked to subsequent IVR modules to handle the error.
- **Interruptible:** Whether to stop playing the prompt as soon as the caller has completed the requested action.
 - Enabled: Stop playing the prompt as soon as the caller has completed the action.
 - Disabled: Continue and finish playing the prompt even if the caller has completed the action.
- **Max Time to Enter:** Number of seconds that the module waits for callers to finish entering the information. If callers exceed the time, the IVR script proceeds to the next module or repeats the current module. The default is 15 seconds.
- **No Input Timeout:** Number of seconds allowed for the caller to provide no input. After the time-out, the NoInput event occurs. If the number of attempts is exceeded, the exception NOT_ENTERED occurs, and the script proceeds to the next module using the No Match branch.
- **Recognition Confidence Threshold:** Percentage confidence level that the system can recognize the caller's input, which can depend on the caller's accent, the quality of the phone connection. This threshold indicates when to consider the speech recognition successful or invalid. The default is 60%. If you set the threshold too high, the number of no matches may be artificially too high.

Confirmation Tab

Prompt used to verify that the system recognized the speech correctly. If the caller confirms the input, the script proceeds to the next module. Otherwise, the prompts are repeated.

- **Required:** Confirmation prompt is played after each caller's input. This option enables all others on this page.
- **If Confidence Level Less Than:** Confirmation prompt played only if the confidence level is low. This option enables all others on this page.
- **Not Required:** Confirmation prompt disabled. This option disables all others on this page.

The screenshot shows the 'Menu Module Properties' dialog box with the 'Confirmation' tab selected. The 'Confirmation' section has three radio buttons: 'Required' (unselected), 'If Confidence Level Less Than' (with a text box containing '75' and a dropdown arrow), and 'Not Required' (selected). The 'Confirmation Dialog' section has two text boxes: 'Max Attempts to Confirm:' (with a value of '3' and a dropdown arrow) and 'No Input Timeout (secs):' (with a value of '3' and a dropdown arrow). The 'Prompt:' section has a text box containing 'ConfirmPromptWithoutVSR' and an 'Edit' button. Below this is a checked checkbox labeled 'Interruptible'. The 'Events:' section contains a table with three columns: 'Event', 'Event's Prompt', and 'Action'.

Event	Event's Prompt	Action
NoMatch	NoMatchPrompt	reprompt
NoInput	NoInputPrompt	reprompt

At the bottom of the dialog are 'Edit', 'Save', and 'Cancel' buttons.

- **Max Attempts to Confirm:** Maximum number of times allowed to enter the confirmation answer. For example, the caller may say 1 or presses 1. If the number of attempts is exceeded, the exception NOT_ENTERED occurs, and the script proceeds to the next module.
- **No Input Timeout:** Number of seconds allowed for the caller to provide no input. After the time-out, the NoInput event occurs.
- **Prompt:** Prompt that describes the confirmation options. The local module variable \$SWI_LITERAL is used to repeat the caller's input as it was recognized.
- **Events:** List of possible confirmation events:
 - NoMatch: Generated if the speech recognition engine cannot find a match in the grammar.
 - NoInput: Generated if the speech recognition engine does not hear any caller input before the time-out.
 - Help: Generated if the caller presses the Help button.

Page Tab

This tab contains the list of Visual IVR prompts that are shown in the customer's browser when a IVR script is executed. You may add images to pages and menus to enhance your brand or to describe a product or service, but you cannot add links to images. Prompts are shown as follows:

- If this tab contains prompts, these are shown whereas prompts in the Prompts tab are not.
- If this tab does not contain prompts, the prompts in the Prompts tab are shown.

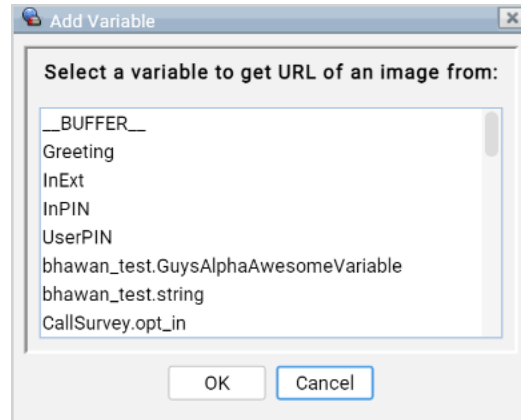
The screenshot shows the 'Menu Module Properties' dialog box with the 'Page' tab selected. The dialog has tabs for General, Branches, Prompts, Events, Confirmation, Page, Text, and Disposition. The 'Page' tab contains two sections: 'Header' and 'Body'. Each section has a table with two columns: 'Type' and 'Text / File Name'. Below each table are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', and 'Convert to ML Prompt'. The 'Add' button for the 'Body' section is open, showing a dropdown menu with options: 'Add File', 'Add File from Variable', 'Add Text', and 'Add Multilingual Prompt'. At the bottom right of the dialog are 'Save' and 'Cancel' buttons.

This tab enables you to add different types of prompts:

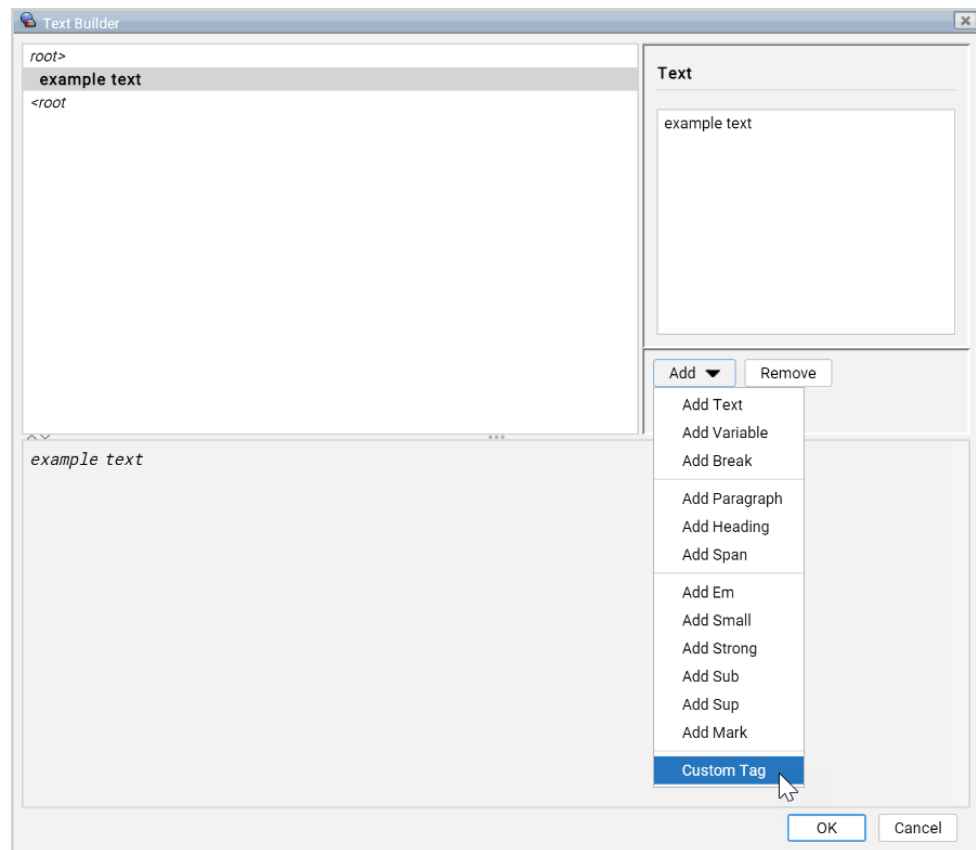
- **File:** URL of an image.

The screenshot shows the 'Add File' dialog box. It has a title bar with a question mark icon and the text 'Add File'. Inside, there is a label 'URL:' followed by a text input field containing the URL 'www.mySite.com/images/five9.png'. At the bottom are 'OK' and 'Cancel' buttons.

- **Text:** Formatted text that can contain variables. You cannot add images. If you select a variable that contains the URL of an image, the URL is displayed.
- **Variable:** Variable that contains the URL of an image.



- **Custom Tag:** See detailed usage information in [Using the Visual IVR URL](#).



The text editor enables you to create HTML text that contains formatted text and variables. You can customize the text with bold, color, and emphasis. Unlike breaks in

the TTS builder, which are associated with speech, breaks in this text editor add a line or horizontal break. The Test button enables you to preview the HTML text.



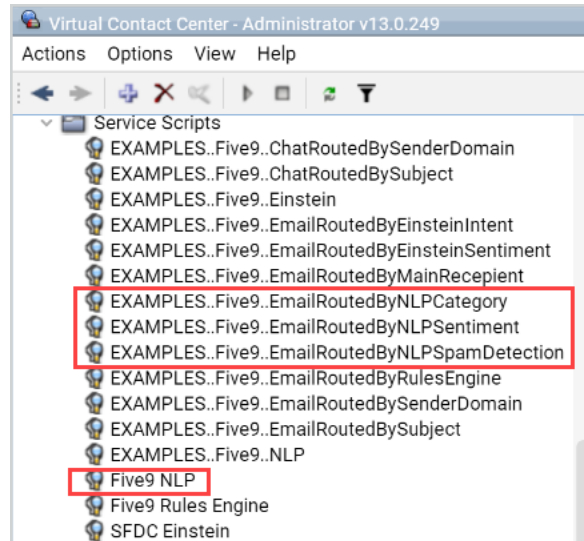
Dispositions Tab

The default disposition is Caller Disconnected. For more information, see [Selecting Dispositions](#).

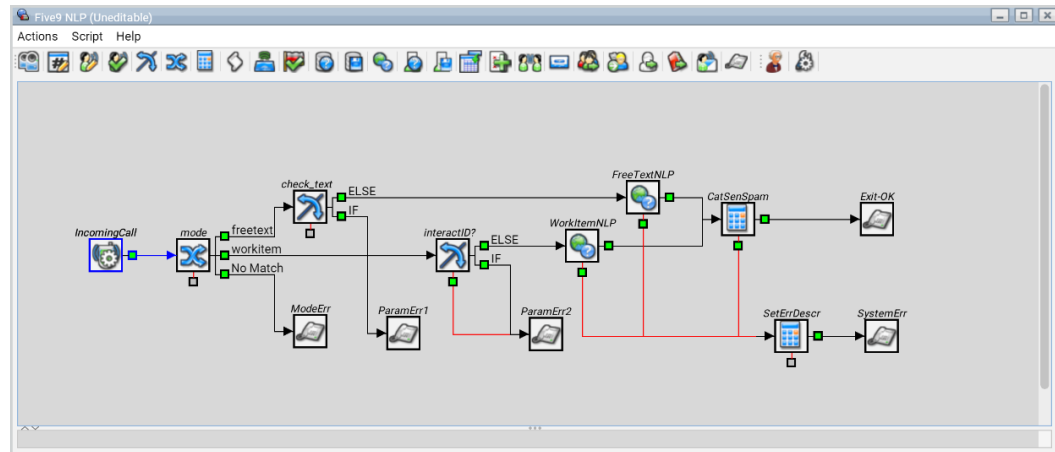
Natural Language Processing Module

The Five9 Natural Language Processing (NLP) module enables you to harness the power of natural language processing in your IVR script. This section details using the Five9 NLP engine to identify sentiment, spam, and sentiment to provide enhanced routing decisions. Using natural language processing requires training the NLP engine for your specific needs. Further information about natural language processing can be found in the [Digital Engagement Administrator's Guide](#).

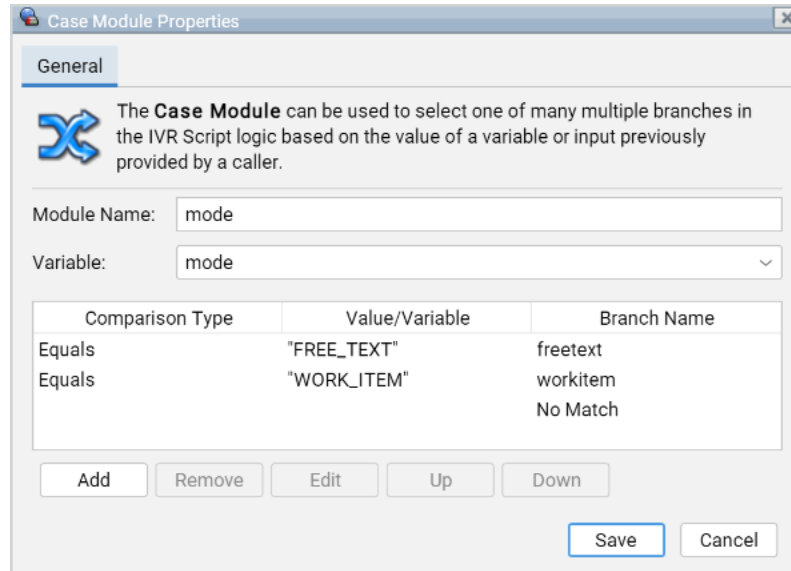
You can access the NLP module for non-editable viewing at **IVR Scripts > Service Scripts > Five9 NLP**. To edit any service script file, right-click the script name and select make a copy, then rename and save or go directly into the IVR builder.



The Five9 NLP module is a pre-packaged IVR script within a module. The example file is shown below.



You can view each module used within the script. For example, opening the properties for the first Case module within the script shows the distribution for free text or work item evaluation.



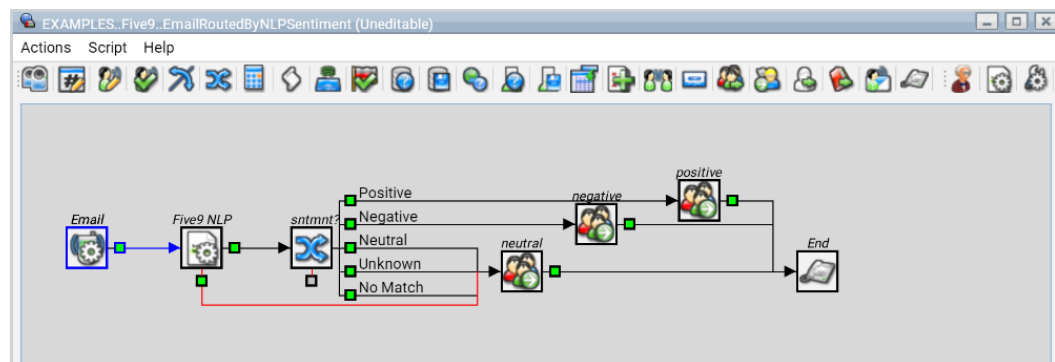
The **Case Module** can be used to select one of many multiple branches in the IVR Script logic based on the value of a variable or input previously provided by a caller.

Module Name:

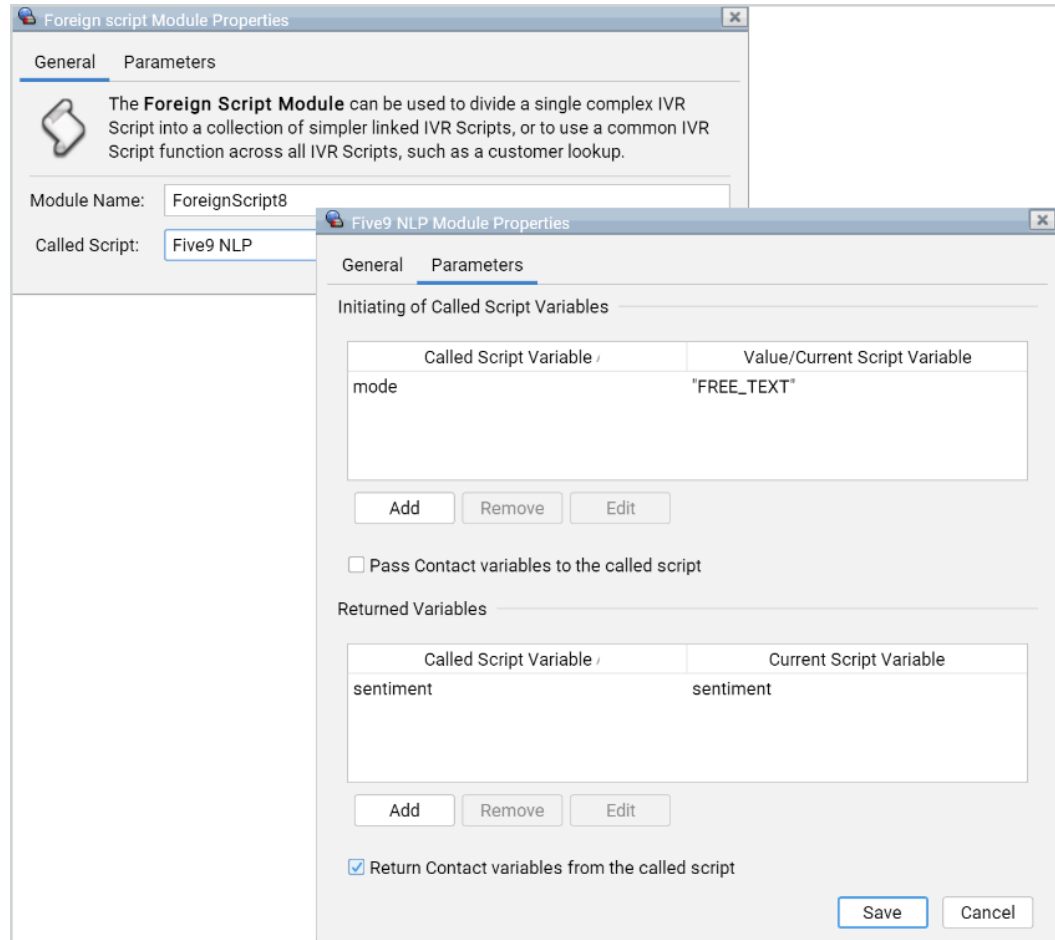
Variable:

Comparison Type	Value/Variable	Branch Name
Equals	"FREE_TEXT"	freetext
Equals	"WORK_ITEM"	workitem
		No Match

You can drag the NLP module directly to your canvas, or create a duplicate of a prepared script and substitute your domain-specific information. The example below shows a script designed to route by NLP sentiment. More information about NLP sentiment, spam detection, and training can be found in the [Digital Engagement Administrator's Guide](#).



Once you have made a duplicate script, or otherwise introduced the NLP module to your IVR script, right-click or double-click to access the module properties.

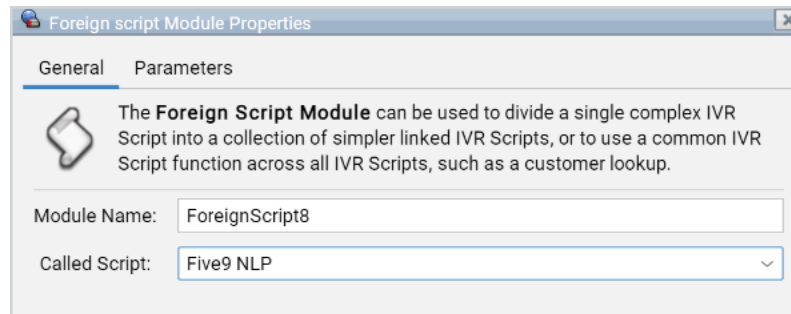


You can rename the module in the **General** tab and assign your specific parameters in the **Parameters** tab.

Properties

[General Tab](#)
[Parameters Tab](#)

General Tab



The screenshot shows the 'Foreign script Module Properties' dialog box with the 'General' tab selected. It contains a description of the Foreign Script Module, a text field for 'Module Name' set to 'ForeignScript8', and a dropdown menu for 'Called Script' set to 'Five9 NLP'.

Foreign script Module Properties

General Parameters

The **Foreign Script Module** can be used to divide a single complex IVR Script into a collection of simpler linked IVR Scripts, or to use a common IVR Script function across all IVR Scripts, such as a customer lookup.

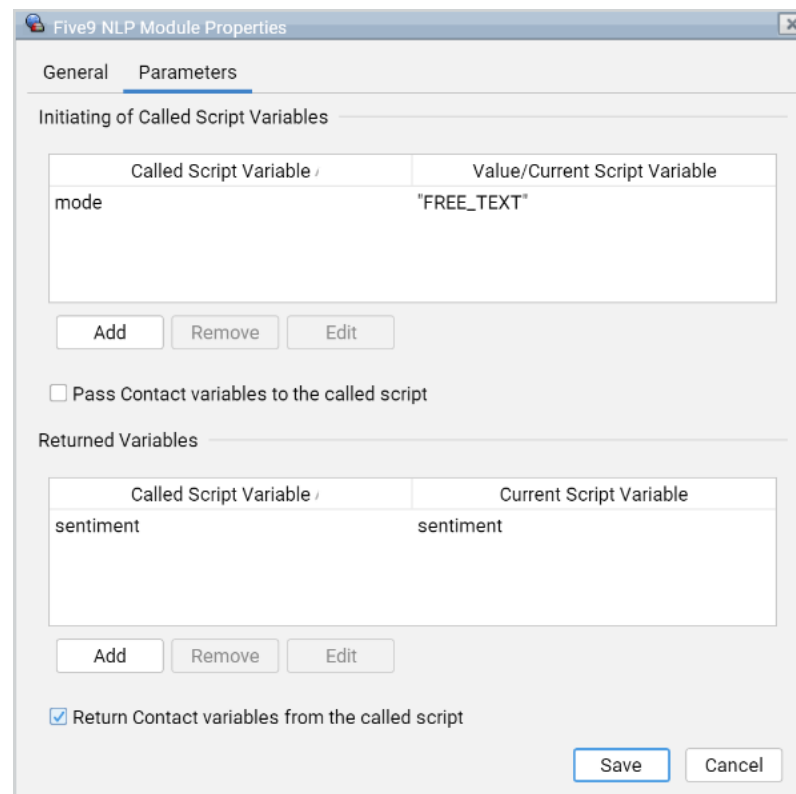
Module Name: ForeignScript8

Called Script: Five9 NLP

- **Module Name:** Name shown in IVR builder.

Parameters Tab

- **Input Parameters:** Contains called script with variable parameters:



The screenshot shows the 'Five9 NLP Module Properties' dialog box with the 'Parameters' tab selected. It displays two tables for managing variables: 'Initiating of Called Script Variables' and 'Returned Variables'. The first table has one entry with 'mode' as the variable and 'FREE_TEXT' as the value. The second table has one entry with 'sentiment' as both the variable and the current script variable. There are 'Add', 'Remove', and 'Edit' buttons for each table, and checkboxes for passing and returning contact variables.

Five9 NLP Module Properties

General Parameters

Initiating of Called Script Variables

Called Script Variable	Value/Current Script Variable
mode	"FREE_TEXT"

Add Remove Edit

☐ Pass Contact variables to the called script

Returned Variables

Called Script Variable	Current Script Variable
sentiment	sentiment

Add Remove Edit

☒ Return Contact variables from the called script

Save Cancel

- **mode:** Determines whether NLP engine evaluates free text or Five9 work item. Default is free text.

- Select check box to pass contact variables to the called script.
- **Possible Errors:**
 - **91:** System errors; incorrect input variables.
 - **92:** Error returned by the NLP system.

Example IVR Scripts with Five9 NLP Module

To see a working example before you create your own, you can open any of the example NLP files listed under the **IVR > Service Scripts** subfolder.

Example

Examples..Five9..EmailRoutedByNLPCategory: Routes an email based on Five9 NLP categorization. Input parameters:

- **category:** Name of the NLP category. This value may be omitted for the main category.
- **Skill2CategoryMapping:** Key-value list {"valueX": "skillX, ..."}, where valueX are the possible values of the category and skillX are the names of corresponding queues for routing.
- **general_skill:** Name of queue for uncategorized emails.

Example

Examples..Five9..EmailRoutedByNLPSentiment: Routes an email based on Five9 NLP sentiment. Input parameters:

- **positive_skill:** Name of a queue used for routing email interactions with positive sentiment.
- **negative_skill:** Name of a queue used for routing email interactions with negative sentiment.
- **general_skill:** Name of a queue used for routing email interactions with neutral or undefined sentiment.

Example

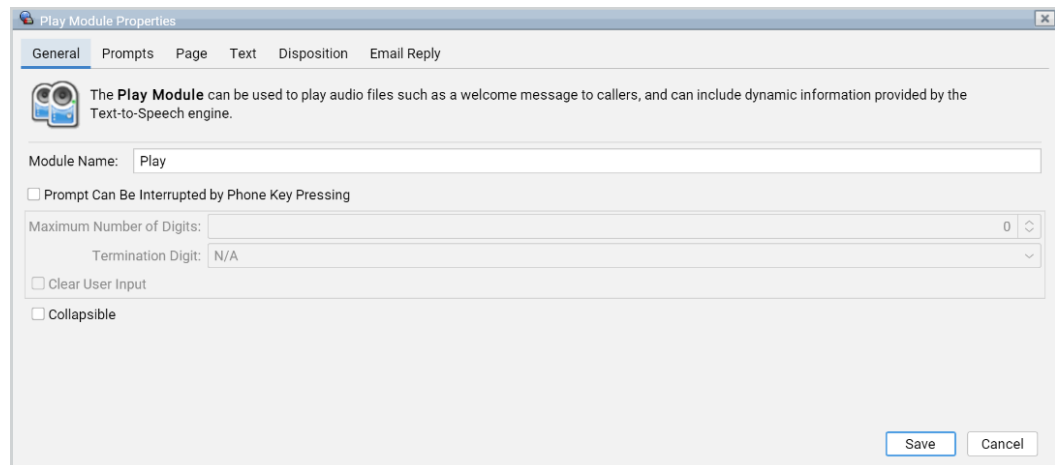
Examples..Five9..EmailRoutedByNLPSpamDetection: Filters out spam based on Five9 NLP categorization. Input parameters:

- **general_skill:** Name of the queue that will receive non-spam emails.

Play Module

The Play module enables you to play prompts and greetings anywhere in the script. For an example, see [Creating an IVR Script](#) and [Using an Agent Extension in an IVR](#).

When using Visual IVR scripts, you may also add images to pages and menus to enhance your brand or to describe a product or service. However, you cannot add links to images. For more information, see [Page Tab](#).



The screenshot shows the 'Play Module Properties' dialog box with the 'General' tab selected. The dialog has tabs for General, Prompts, Page, Text, Disposition, and Email Reply. The General tab contains a description of the Play Module, a text field for 'Module Name' (set to 'Play'), a checkbox for 'Prompt Can Be Interrupted by Phone Key Pressing', a 'Maximum Number of Digits' spinner (set to 0), a 'Termination Digit' dropdown (set to N/A), and checkboxes for 'Clear User Input' and 'Collapsible'. 'Save' and 'Cancel' buttons are at the bottom right.

Properties

[General Tab](#)
[Prompts Tab](#)
[Page Tab](#)
[Dispositions Tab](#)
[Email Reply Tab](#)

General Tab

- **Module Name:** Name that represents the role of the module in the IVR script. In Visual IVR scripts, the name is converted to text that appears in the heading of the IVR window.
- **Prompt can be interrupted by phone key pressing:** Select this option when your prompt asks for input from the caller, such as *press 1 for sales* or when you want to allow a termination digit. When using this option, follow the Play module with

the Get Digits or Input module.

Note

When you expect caller input, Five9 recommends that you use the Get Digits module instead of the Play module.

This option activates the other options below.

- **Maximum Number of Digits:** Available only when using interruptible prompts. Represents the number of digits that the caller may press. When the digits are entered, the prompt stops, and the script proceeds to the next module. For example, if your prompt is *press 1 for sales, 2 for Support, or 3 for Billing*, the maximum number of digits is 1. If you are asking for a four-digit extension, the maximum number of digits is 4.
- **Termination Digit:** Option for callers to exit the module without entering an extension by pressing the key specified.
 - Enabled: the possible options are # (pound) and * (star).
 - Disabled: the value N/A disables the option.
- **Clear User Input:** Whether to erase the information entered by the caller before reaching this module.
 - Enabled: delete all user input before this module.
 - Disabled: Do not enable if, before this module, you have a prompt that provides instructions, such as pressing a key (termination digit) to exit the queue.
- **Collapsible:** Visual IVR feature that applies to all types of prompts. Whether to concatenate all the prompts in the module whether they are played (standard prompts) or displayed (Visual IVR prompts) to callers.
 - Enabled: Concatenate prompts. Callers see at once all the prompts in the same window.
 - Disabled: Do not concatenate prompts. Callers hear or see each prompt after performing the requested action.

Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). Each prompt can be interruptible.

In Visual IVR scripts, TTS prompts are converted to text that appears in the body of the IVR window.

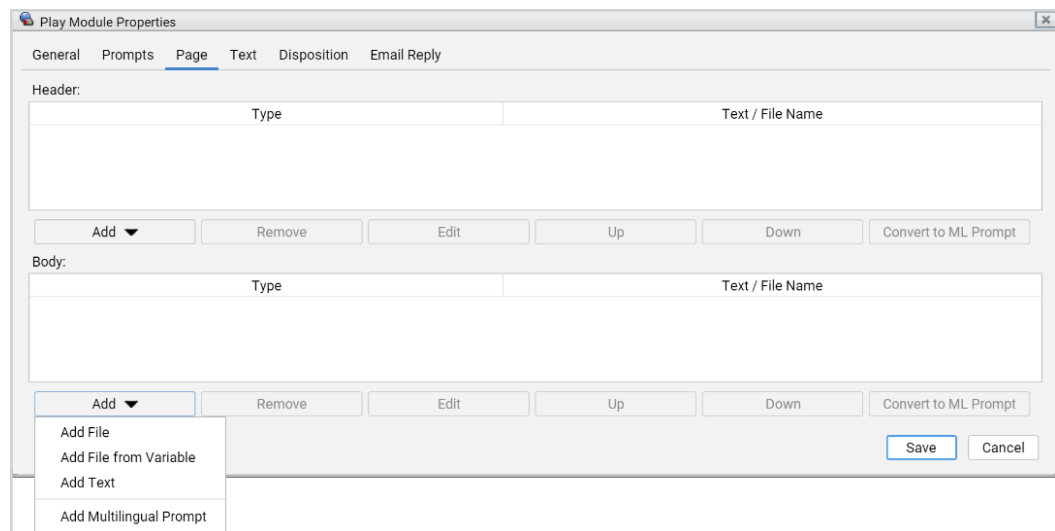
- **TTS Say As Options:** Described in the table following this one.

Page Tab

This tab contains the list of Visual IVR prompts that are shown visually in the customer's Web browser when a IVR script is executed.

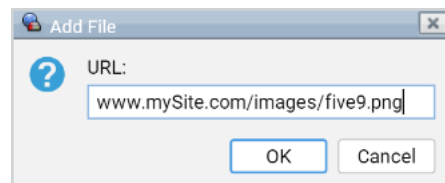
If this tab contains prompts, these are shown whereas prompts in the Prompts tab are not.

If this tab does not contain prompts, the prompts in the Prompts tab are shown.

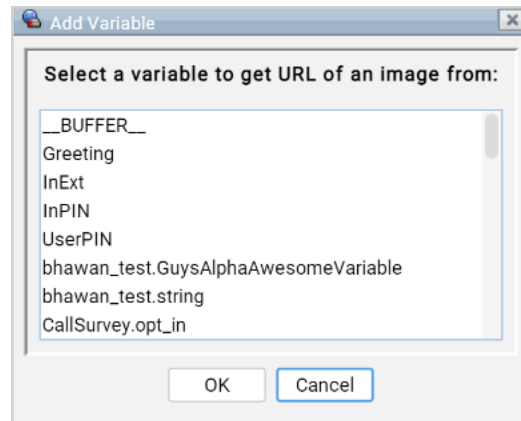


This tab enables you to add different types of prompts:

- **File:** URL of an image.

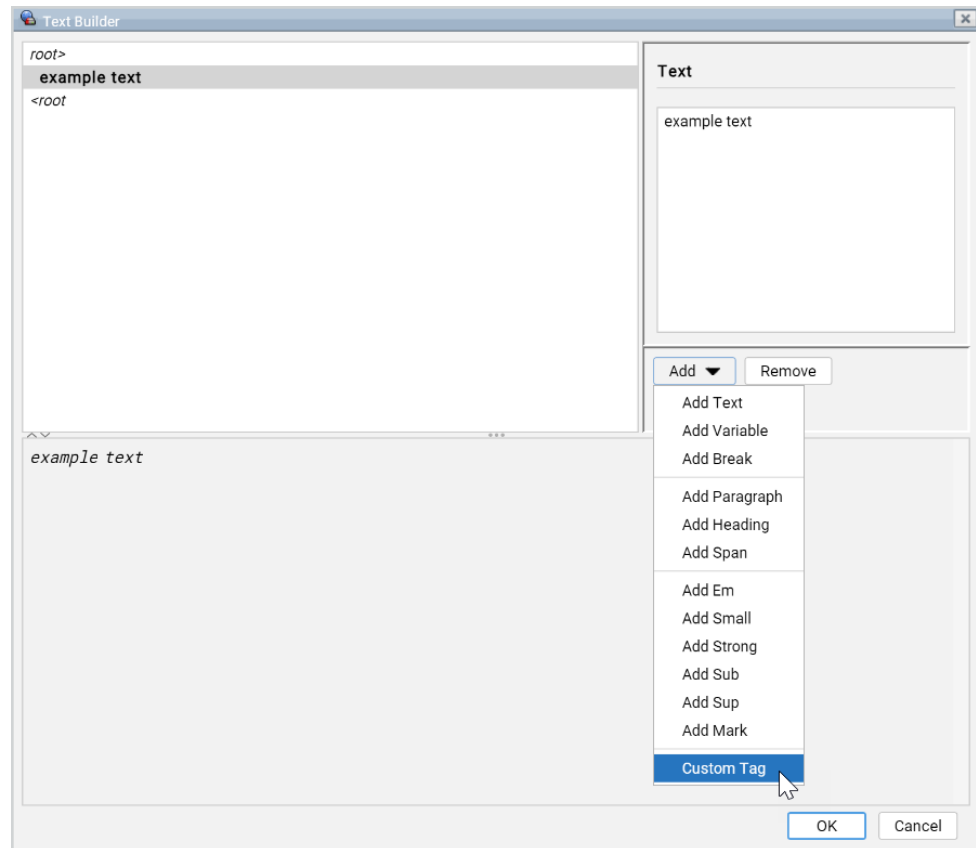


- **Variable:** Variable that contains the URL of an image.



- **Text:** Formatted text that can contain variables. You cannot add images. If you select a variable that contains the URL of an image, the URL is displayed.

- Custom Tag: For detailed usage information, see [Using the Visual IVR URL](#).



The text editor enables you to create HTML text that contains formatted text and variables. You can customize the text with bold, color, and emphasis. Unlike breaks in the TTS builder, which are associated with speech, breaks in this text editor add a line or horizontal break. The Test button enables you to preview the HTML text.



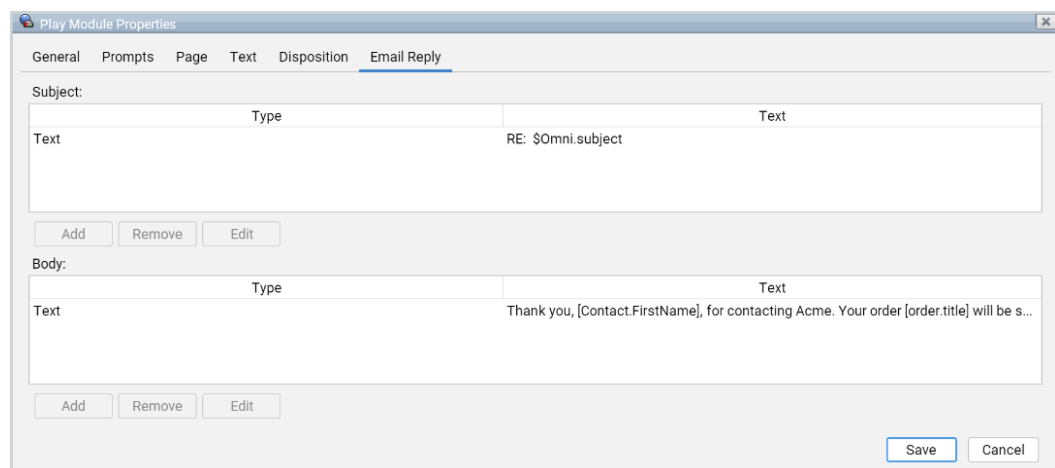
Dispositions Tab

The default disposition is Caller Disconnected. For more information, see [Selecting Dispositions](#).

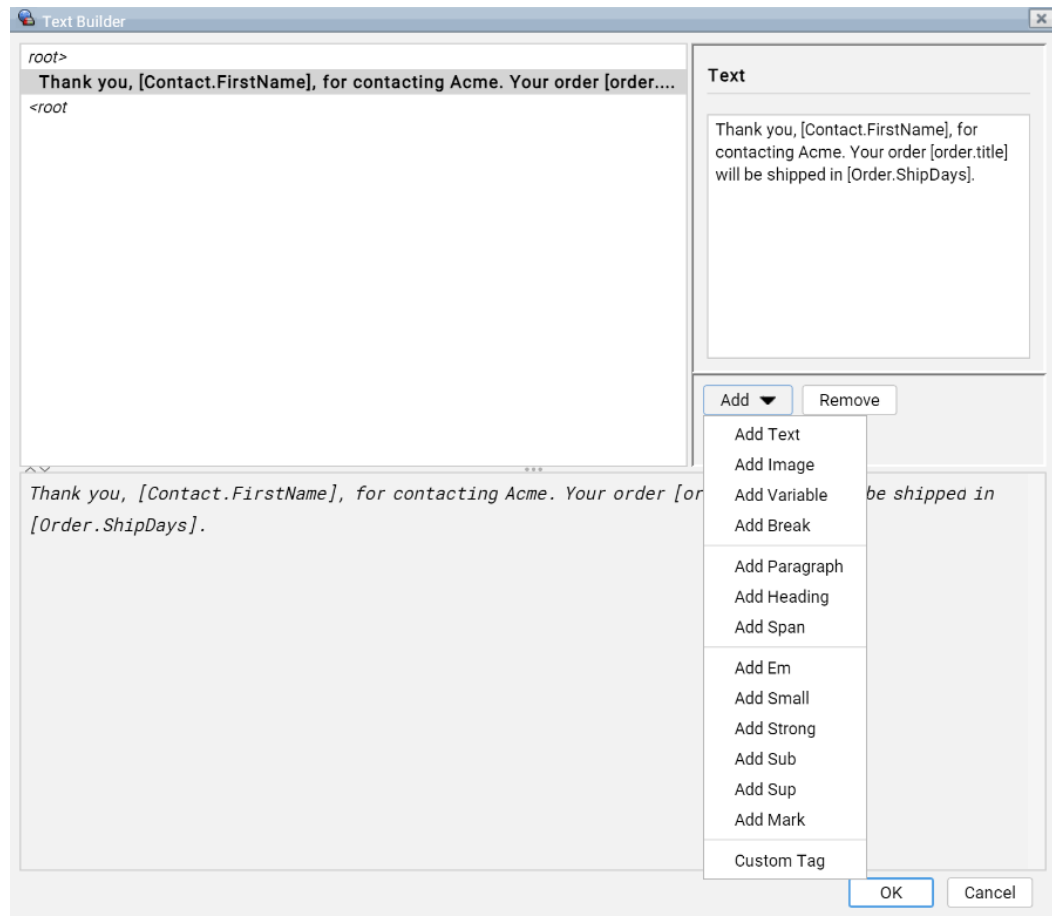
Email Reply Tab

Use this tab with Five9 Engagement Workflow to create a rich text automatic response or by using the IVR Text Builder to send an acknowledgment. The reply subject is pre-populated and editable.

You can specify a single reply for each Play module that you use and configure.



To create the body of your reply using the IVR Text Builder, click **Add**.

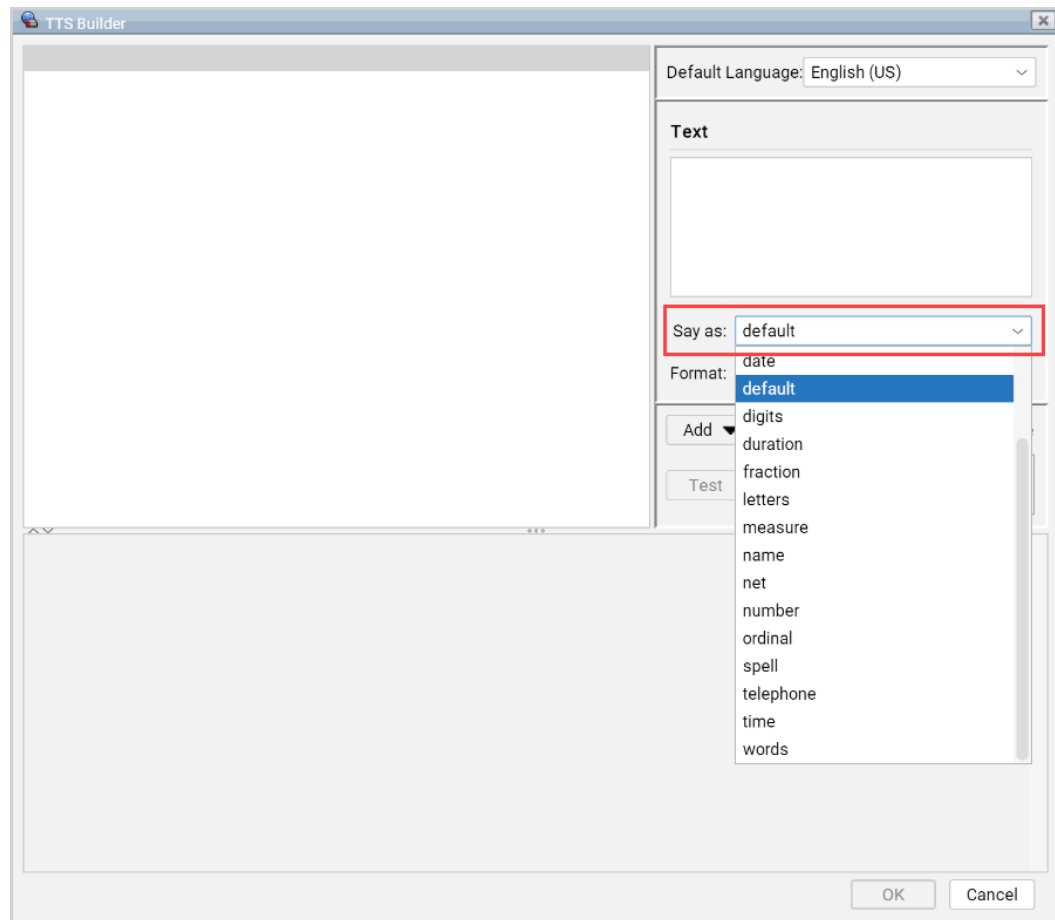


- **Type:** Reply format.
- **Text:** The email body of your reply may include up to 24KB in the form of: HTML, call variables, and embedded images.

Example

Thank you, [Contact.FirstName], for contacting Acme. Your order [order.title] will be shipped in [Order.ShipDays].

Text-to-Speech *Say As* Options



The Text-to-Speech (TTS) ***Say as*** options enable you to play words or phrases to your customers in a contextual format. In certain instances, the text should be split apart if it contains words that require special treatment. For example, phone numbers that are part of a larger phrase should be split out into a separate text element so that you can apply the ***Say as telephone*** rule to it. The descriptions and data types are detailed below.

Say as...	Format	Description	Data Type
Acronym	N/A	Sequences of letters and digits are spoken as words when this is considered natural in the target language, such as NATO or UNESCO for English. Otherwise, letters	STRING

Say as...	Format	Description	Data Type
		and digits are pronounced individually, like API for English.	
Address	N/A	Used for postal addresses.	STRING
		Note: Regional localization for Languages improves accuracy in speaking addresses. For example, English (GB) is better able to accurately reference United Kingdom addresses than English (US).	
Cardinal	N/A	Supported if relevant in the target language. Roman cardinals are often supported.	INTEGER, STRING
Currency	N/A	For text that is a currency amount the currency symbol may be present in the enclosed text. This mode supports currencies commonly specified in the country target language, such as \$ and ¢ for US English.	CURRENCY, NUMBER, INTEGER, STRING
Date	d, dm, dmy, m, md, mdy, my, y, ym, ymd	Use for date strings.	DATE, STRING
Decimal	N/A	Same as number with format decimal. Use of the separator for the integral part is optional. For example, 123456.123 and 123,456.123 are pronounced in the same way for US English.	INTEGER, NUMBER
Digits	N/A	Numbers must be read digit by digit. Decimal periods or commas should be pronounced. Same as number with format digits.	NUMBER, INTEGER
Duration	h, hm, hms, m, ms, s	Duration with format hms is read as <h> hour(s), <m> minute(s), and <s> seconds.	INTEGER, TIME, STRING
Fraction	N/A	Same as number; with format fraction. This mode will pronounce 1/3 as one	STRING

Say as...	Format	Description	Data Type
		third.	
Letters	N/A	Pronounce alphanumerical strings as a sequence of individual letters and digits. For true spelling of all readable characters, use the Say-as value spell.	STRING
Measure	N/A	Variety of units, like km, hr, dB, lb, MHz are supported; the units may appear immediately next to a number, such as 1cm, or be separated by a space, as in 15 ms.	STRING
Name	N/A	Interpret a string as a proper name if possible.	STRING
Net	email, uri	May be used for email addresses.	STRING
Number	Cardinal, decimal, digits, fraction, ordinal, telephone	Only formats relevant in the target language are supported. All the format values are supported as interpret-as values as well, behaving the same for either syntax.	INTEGER, NUMBER, STRING
Ordinal	N/A	Only formats relevant in the target language are supported. This mode will adhere to specified rank or series.	INTEGER
Spell	N/A	Characters in the text string are pronounced as individual characters.	STRING
Telephone	N/A	Supports telephone numbers commonly specified in the country corresponding to the target language.	INTEGER, STRING
Time	h, hm, hms	Hour should be less than 24, minutes and seconds less than 60. AM/PM is read only if explicitly specified.	TIME, STRING
Words	N/A	Words are read instead of pronouncing them as strings of individual letters and digits. The characters of a word may still be uttered individually for particularly difficult to pronounce character sequences. This mode is also meant for acronyms to be read as words.	STRING

Query Module

The Query module enables you to obtain database information from an external Web site, access third-party Web services, or integrate an external source of contact data. You can send GET or POST requests in which input parameters are appended to the request URL or XML Web service POST requests.

When using Visual IVR scripts, you may also add images to pages and menus to enhance your brand or to describe a product or service. However, you cannot add links to images. For more information, see [Page Tab](#).

[Types of Database](#)
[Properties](#)
[Use Case](#)

This chapter is intended for developers who understand these technologies and protocols:

- Client-server architecture
- XML and Web Services
- HTTP protocol
- Web programming languages, such as ASP, Perl, PHP, Java, or Ruby

Types of Database

The flow of information differs depending on the type of database that you access:

- If the database supports Web APIs, you can access information directly with the Query module.
- If the database, such as MySQL™, does not support Web APIs, you can use a Web server as a proxy for the database:
 - a Send an HTTP request from the Query module to your Web server. To create a secure connection, use basic access authentication.
 - b Add a script to your Web server. The script returns the data in the format specified in the script.

Properties

The URL standard contains a port specification, for example: `www.domain.com` means `http://www.domain.com:80`. The default port is generally omitted. Five9 URLs use only port 80.

[General Tab](#)
[Request Tab](#)
[Response Tab](#)
[Prompts Tab](#)
[Page Tab](#)
[Text Tab](#)

General Tab

The screenshot shows the 'Query Module Properties' dialog box with the 'General' tab selected. The dialog has a title bar with a close button. Inside, there's a tabbed interface with 'General', 'Request', 'Responses', 'Prompts', 'Page', and 'Text'. The 'General' tab contains a description: 'The Query Module can be used to look up or post information maintained by an external web server.' Below this, there are fields for 'Module Name' (containing 'Query2') and 'Base URL path'. A table with two columns, 'URL Part' and 'Comment', is present, with buttons 'Add', 'Remove', 'Edit', 'Up', and 'Down' below it. At the bottom, there are fields for 'Method' (set to 'GET'), 'Timeout (sec)' (set to '5'), and a checkbox for 'Save Number of Records to:'. 'Save' and 'Cancel' buttons are at the bottom right.

URL Part	Comment
----------	---------

- **Base URL Path:** HTTP or HTTPS request URL. For secure connections, use certificates when the external Web server uses a root certificate (unsigned/self-signed public key certificate). You can import CRT files or enable Accept All Certificates. To configure certificates go to **Actions > Configure > Certificates**.

Important: If your parameters contain sensitive data, use secure URLs and the POST method. You cannot save a query that contains sensitive data with a non-secure URL or the GET method.

- **URL Part:** Specify portions of the intended URL by using variables to form either part or all of the URL path. The URL should have this format:

```
<BaseURL>/<URLPart1>/.../<URLPartN>/
```

GET parameters should have this format:

```
<BaseURL>/<URLPart1>/.../<URLPartN>/?<param1>=  
=<value1>&...&<paramM>=<valueM>
```

- **Method:** Request method:
 - GET
 - POST
 - PUT
 - DELETE
- **Fetch Timeout:** Maximum number of seconds for receiving a response from the script. The default is 5 seconds. After the time-out, the script exits the module.
- **Save Number of Records to:** If several records match the search criteria, whether to save the number of matched records in the selected integer CAV variable.

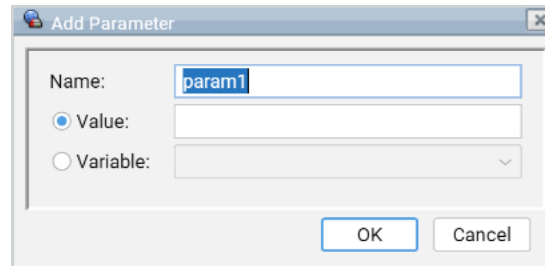
Request Tab

Contains a query string parameter area to define part(s) of the URL containing data that does not fit conveniently into a hierarchical path structure:

The screenshot shows the 'Query Module Properties' dialog box with the 'Request' tab selected. The dialog has five tabs: General, Request, Responses, Prompts, and Text. The 'Request' tab contains the following elements:

- A table with two columns: 'HTTP Header' and 'Value / Variable'.
- Buttons: 'Add', 'Remove', and 'Edit'.
- An 'Authorization Profile' section with a checkbox and two radio buttons: 'Name' (selected) and 'Variable'.
- A section for 'form-urlencoded', 'form-data', and 'raw' with 'form-urlencoded' selected.
- A table with two columns: 'Parameter' and 'Value / Variable'.
- Buttons: 'Add', 'Remove', and 'Edit'.
- 'Save' and 'Cancel' buttons at the bottom right.

- **Parameters:** You can import a file that contains your request code, and you can add variables. All parameters are encoded and separated by an ampersand (&). Supports `multipart/form-data`.

A dialog box titled "Add Parameter" with a close button (X) in the top right corner. It contains a "Name:" label followed by a text input field containing "param1". Below this are two radio buttons: "Value:" (selected) and "Variable:". The "Variable:" option is followed by a dropdown menu. At the bottom right are "OK" and "Cancel" buttons.

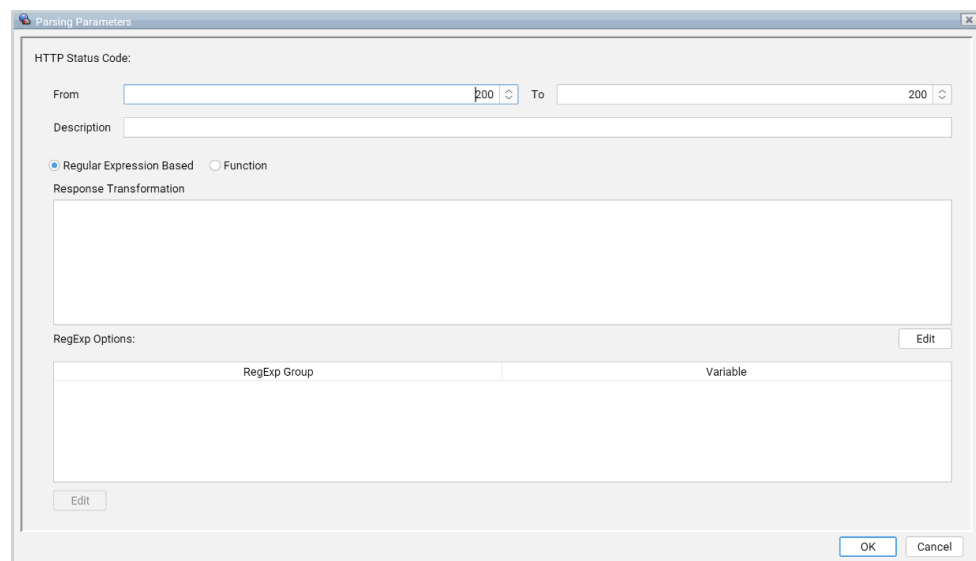
- **Authorization Profile:** Name of authorization profile for validation.

Response Tab

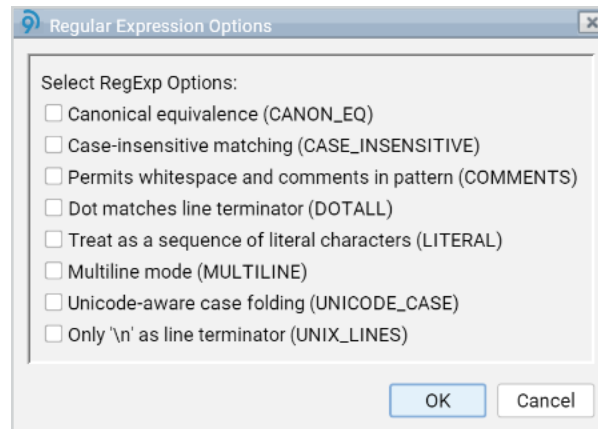
The Response tab enables you to configure several sets of parsing parameters. The parsing parameters correspond to one or more http status codes or their ranges. Route the call to the appropriate following branch:

- Success: Data retrieved successfully.
- Timeout: Query exceeded time allowed.
- Failure: No records found.
- **HTTP Parsing:** Add RegEx or function to parse the returned parameters.

Returned HTTP Status Code (INT) and Reason Phrase (STRING) may be saved in variables.

A dialog box titled "Parsing Parameters" with a close button (X) in the top right corner. It contains a section for "HTTP Status Code:" with "From" and "To" input fields, both containing "200". Below this is a "Description" text area. There are two radio buttons: "Regular Expression Based" (selected) and "Function". Below these is a "Response Transformation" text area. At the bottom, there is a section for "RegExp Options:" with an "Edit" button. Below this is a table with two columns: "RegExp Group" and "Variable". At the bottom right are "OK" and "Cancel" buttons.

- **Regular Expression Based:**



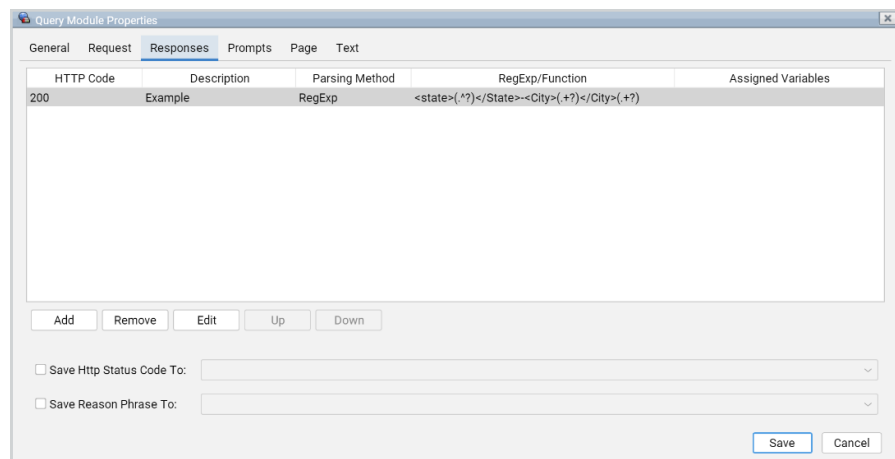
- Response transformation field: Add regular expressions enclosed in parentheses. Regular expressions are automatically added to the RegExp Group field.

Example

```
<State>(.*?)</State>-<City>(.*?)</City>(.*?)
```

This expression matches the first instance of one or more characters, stops matching, and returns the result.

- Map the expressions to IVR variables in the RegExp Group field. Assign the result to a variable. In this case, the result is assigned to the `City` variable.



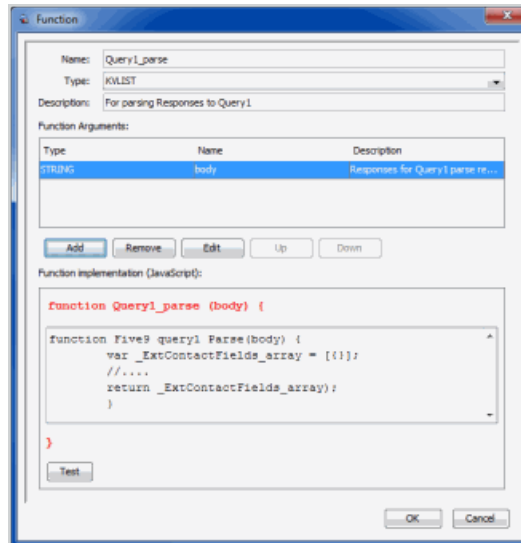
- **From** and **To** fields contain 3-digit integers.

- Parsing rules may overlap by code ranges. In this instance, the top rule applies.
 - Rule order may be changed using Up/Down buttons.
 - If the status code of a response matches a rule, the main exit of the Query module is activated. Otherwise, an exception occurs.
- **Function:** Enables you to apply any function for parsing the query response.

The screenshot shows the 'Parsing Parameters' dialog box. It has a title bar with a close button. The main area is divided into sections. The 'HTTP Status Code' section has 'From' and 'To' dropdown menus, both set to '200'. Below this is a 'Description' text field. There are two radio buttons: 'Regular Expression Based' (unselected) and 'Function' (selected). Under the 'Function' section, there are fields for 'Name', 'Type', and 'Description'. A 'New...' button is located to the right of the 'Name' field. Below these is a large text area for 'Function Arguments'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

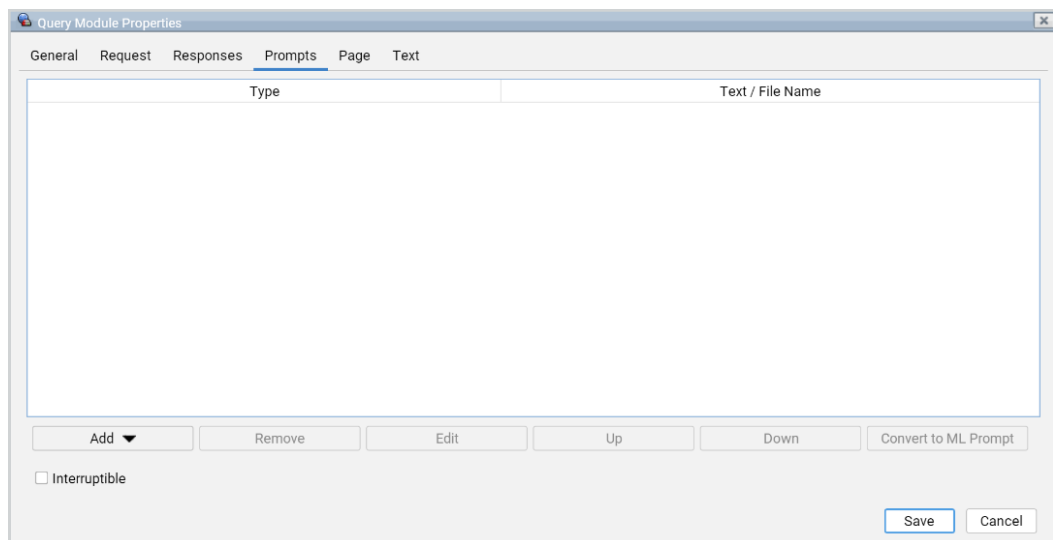
- Variable
 - Name: Variable source, such as `__BUFFER__`, or `__ExtContactFields__`.
 - Type: Variable type, such as: String, KVList.
 - Description: Variable description.
- Function: Use canned functions or define a new function and (JavaScript) implementation.
 - Name: Defined functions, such as: concat, substring, ...
 - Description: Function description.
 - New (function): Enables function creation.

New Function Example



Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). Each prompt can be interruptible.



- **Interruptible:** Whether to stop playing the prompt as soon as the requested operation is completed.
 - Enabled: Stop playing the prompt as soon as the caller has completed the action.
 - Disabled: Finish playing the prompt even if the caller has completed the action.

Page Tab

This tab contains the list of Visual IVR prompts that are shown visually in the customer's Web browser when a IVR script is executed.

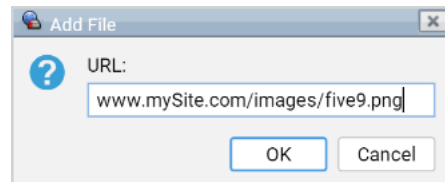
If this tab contains prompts, these are shown whereas prompts in the Prompts tab are not.

If this tab does not contain prompts, the prompts in the Prompts tab are shown.

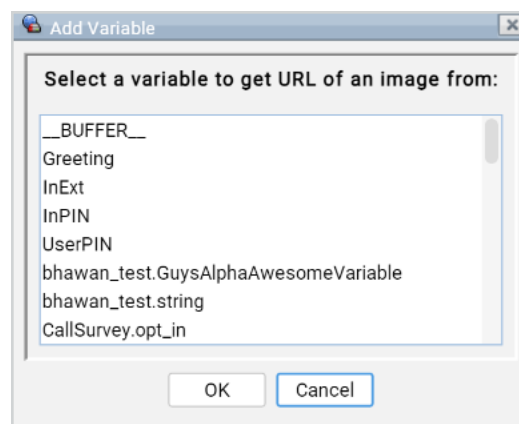
The screenshot shows the 'Query Module Properties' dialog box with the 'Page' tab selected. The dialog has tabs for General, Request, Responses, Prompts, Page, and Text. The 'Page' tab contains two sections: 'Header' and 'Body'. Each section has a table with two columns: 'Type' and 'Text / File Name'. Below each table are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', and 'Convert to ML Prompt'. The 'Add' button in the 'Body' section is open, showing a dropdown menu with options: 'Add File', 'Add File from Variable', 'Add Text', and 'Add Multilingual Prompt'. At the bottom right of the dialog are 'Save' and 'Cancel' buttons.

This tab enables you to add different types of prompts:

- **File:** URL of an image.



- **Variable:** Variable that contains the URL of an image.



- **Text:** Formatted text that can contain variables. You cannot add images. If you select a variable that contains the URL of an image, the URL is displayed.

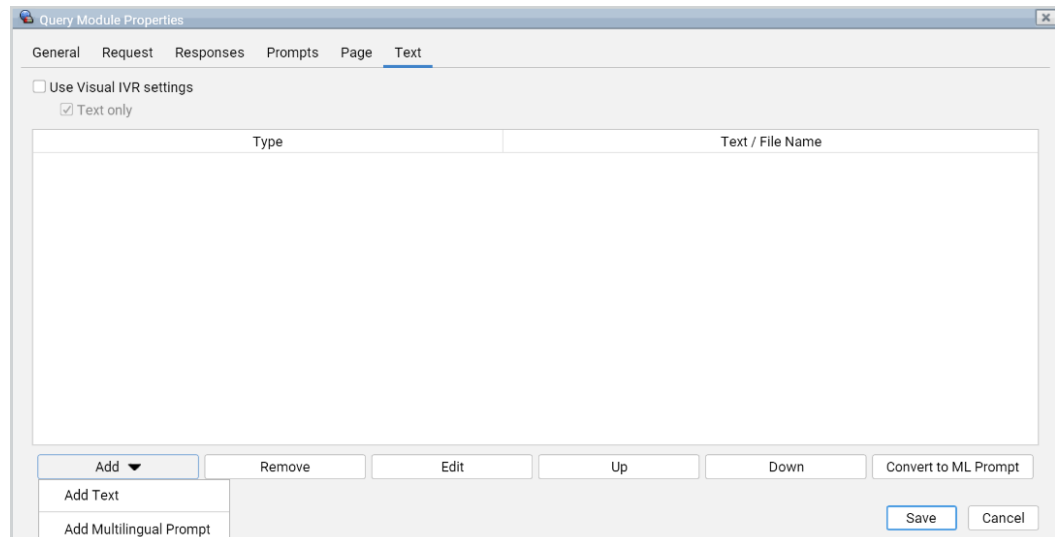
The text editor enables you to create HTML text that contains formatted text and variables. You can customize the text with bold, color, and emphasis. Unlike breaks in the TTS builder, which are associated with speech, breaks in this text editor add a line or horizontal break.

The Test button enables you to preview the HTML text.



Text Tab

This tab enables configuration options for chat, SMS, email, and integration channels.



- **Use Visual IVR Settings:**

- Enabled: digital engagement interactions receive the data configured in the Page tab.

Text only: If enabled, the XML/HTML are removed from the data received from in the Page tab configuration.

- Disabled:

- Text only box is disabled.
- The grid and grid buttons enable configuration of a separate prompt for digital engagement interactions.
- The same items can be added as for Page/Body.
- Add Text: Text and variables can be added. The same control as is used for a Query/POST request may be used here.
- Add Multilingual Prompt: Only multilingual prompts with the type TEXT may be used here.

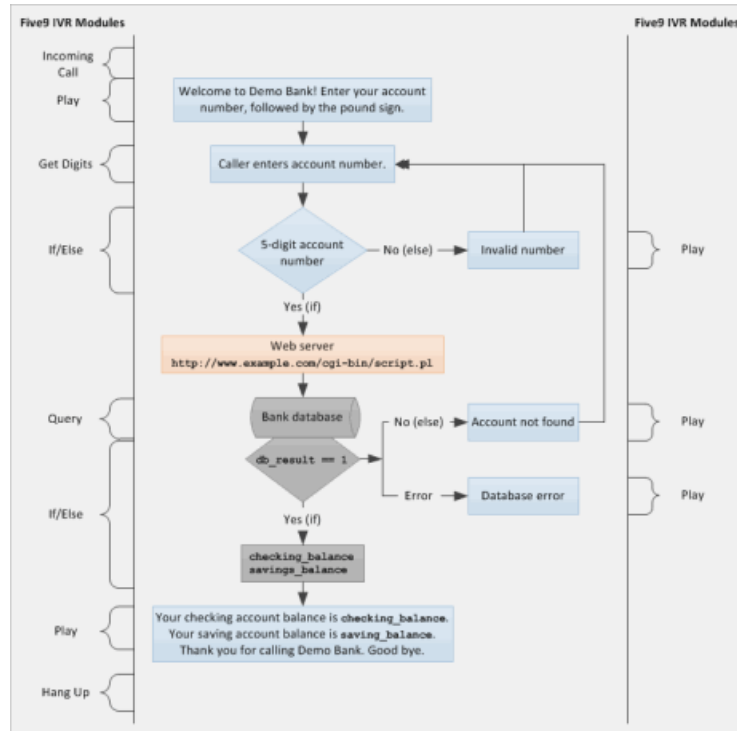
Use Case

If you develop and maintain the database and all the code necessary to access the database, import [banking.five9ivr](#) into the IVR workspace to see a working example.

This example shows how to query a database that you develop and maintain.

[Preparing Your Database](#)
[Creating Script Variables](#)
[Configuring the Query Module](#)
[Customizing the Script](#)
[Testing Your Implementation](#)

This figure shows the flow of information between the IVR modules, your Web server, and the database when a customer calls to obtain account balances.



Preparing Your Database

Define the variables for the request, the response, and the Query module.

Create a database table with these columns:

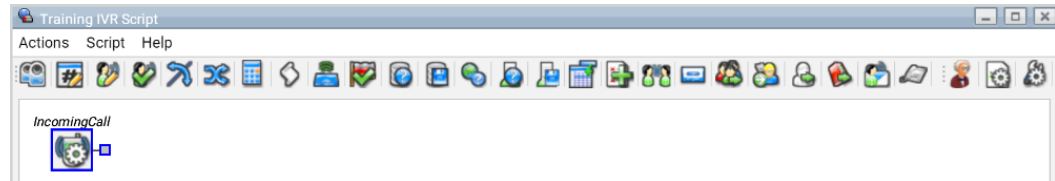
- `account_number`: 5-digit number provided by the customer.
- `checking_balance`: account balance and currency returned by the search.
- `savings_balance`: account balance and currency returned by the search.

Example

```
CREATE TABLE accounts (
  account_number INTEGER(5),
  checking_balance DECIMAL(10,2),
  saving_balance DECIMAL(10,2)
)
```

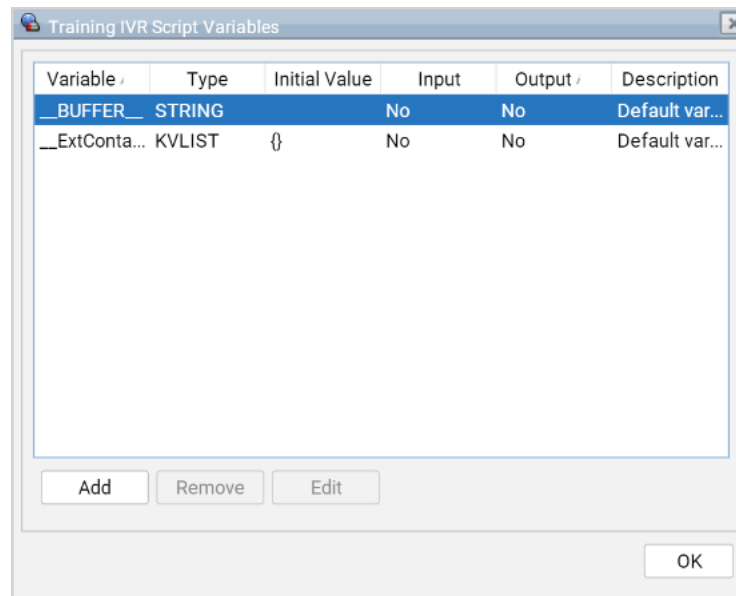
Creating Script Variables

Prepare the variables that you will use in the query. In this case, `account_number` is the only variable.

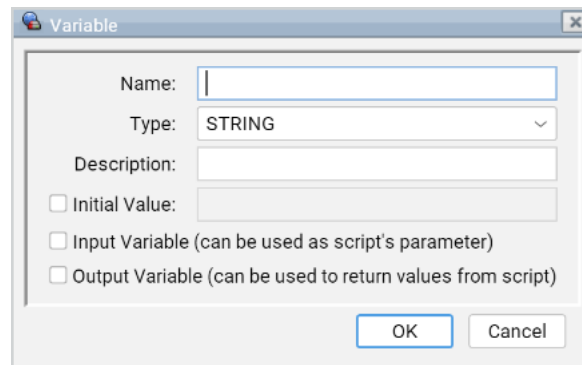


- 1 In the IVR workspace, select **Script > Variables**.

The Variables window contains a default variable that you cannot modify or delete.



- 2 To start creating variables, click **Add**.

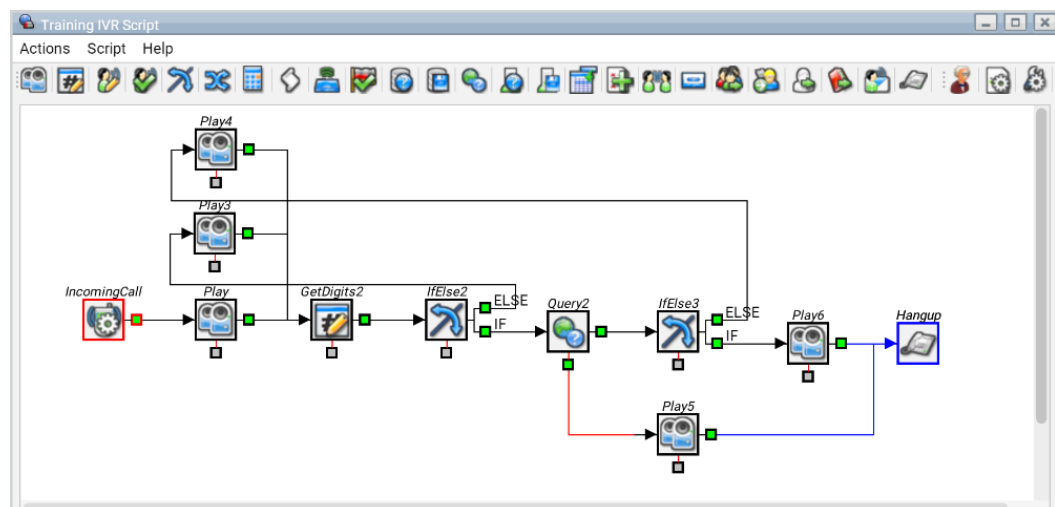


- 3 Name and describe the variable.
- 4 Click **OK**.
- 5 When done creating variables, click **OK**.

Configuring the Query Module

[General Tab](#)
[Request Tab](#)
[Response Tab](#)
[Prompts Tab](#)
[Page Tab](#)
[Text Tab](#)

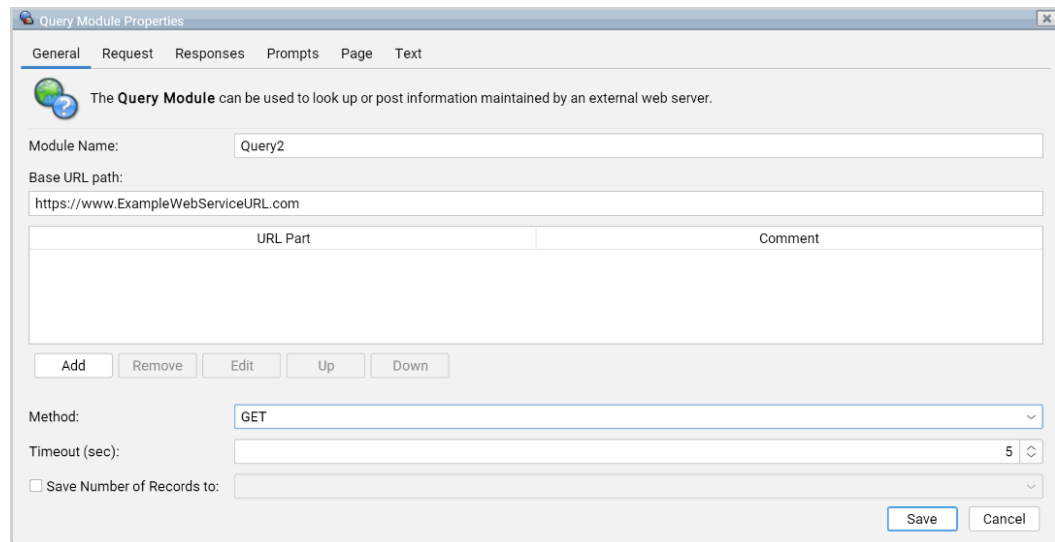
The modules in this figure correspond to those in the flowchart above. The Query module is highlighted in red. Of the modules shown, we describe only how to configure the Query module.



To open the properties window, double-click the module.

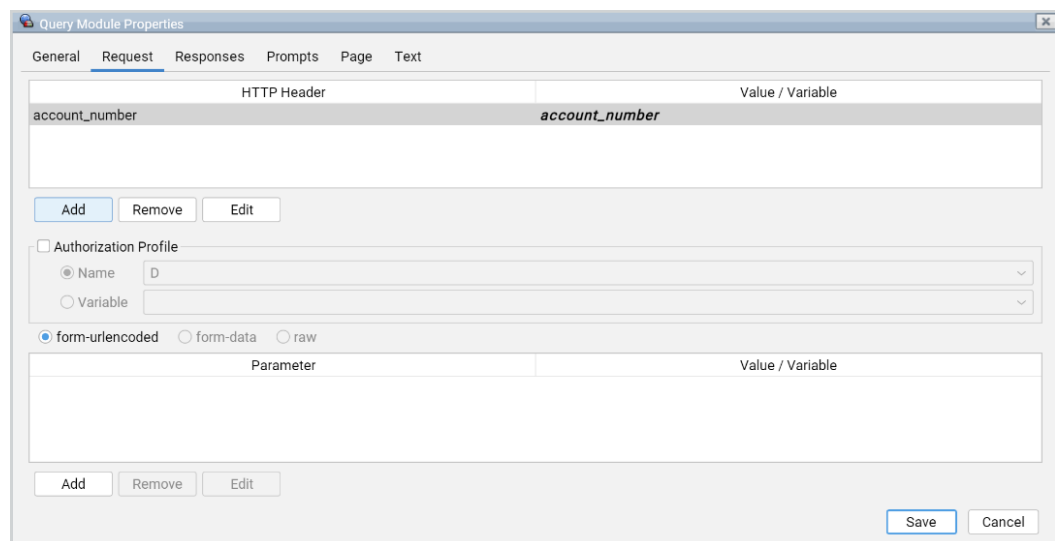
General tab. Enter basic information about the module:

- **URL:** Request URL of the Web server. You can use a secure TLS connection.
- **Method:** URL-encoded GET



The **Query Module Properties** dialog box is shown with the **General** tab selected. The dialog includes tabs for General, Request, Responses, Prompts, Page, and Text. A description states: "The **Query Module** can be used to look up or post information maintained by an external web server." The **Module Name** is "Query2". The **Base URL path** is "https://www.ExampleWebServiceURL.com". Below this is a table with columns "URL Part" and "Comment". At the bottom, the **Method** is set to "GET", the **Timeout (sec)** is "5", and there is an unchecked checkbox for "Save Number of Records to:". **Save** and **Cancel** buttons are at the bottom right.

Request tab. Enter the input variables as parameters and the HTTP headers that you want. In this case, the only variable is the account number.

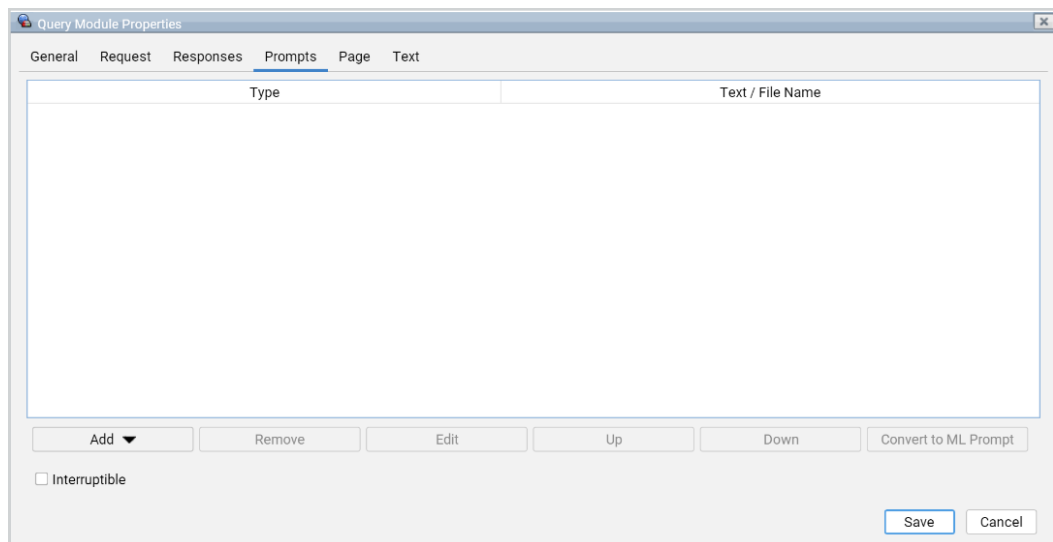


The **Query Module Properties** dialog box is shown with the **Request** tab selected. It features a table for "HTTP Header" with columns "HTTP Header" and "Value / Variable". One entry is shown: "account_number" with value "account_number". Below the table are **Add**, **Remove**, and **Edit** buttons. An **Authorization Profile** section has radio buttons for "Name" (selected, with value "D") and "Variable". Below this are radio buttons for "form-urlencoded" (selected), "form-data", and "raw". At the bottom is a table for "Parameter" with columns "Parameter" and "Value / Variable", and **Add**, **Remove**, and **Edit** buttons. **Save** and **Cancel** buttons are at the bottom right.

Response tab. Select XML Element Attributes, and enter these parameters:

Parameter	Variable
checking	checking_balance
DB_result	DB_result
saving	saving_balance

Prompts tab. This section is optional. Instead you can configure a Play module to play a prompt while the customer is waiting, or you can omit playing a prompt.



The image shows the 'Query Module Properties' dialog box with the 'Prompts' tab selected. The dialog has tabs for General, Request, Responses, Prompts, Page, and Text. The Prompts tab contains a table with two columns: 'Type' and 'Text / File Name'. Below the table are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', and 'Convert to ML Prompt'. There is also a checkbox for 'Interruptible' and 'Save'/'Cancel' buttons at the bottom right.

Type	Text / File Name
------	------------------

Buttons: Add, Remove, Edit, Up, Down, Convert to ML Prompt

☐ Interruptible

Buttons: Save, Cancel

Page tab. Ignore this tab.

The screenshot shows the 'Query Module Properties' dialog box with the 'Page' tab selected. The dialog has tabs for General, Request, Responses, Prompts, Page, and Text. The 'Page' tab contains two sections: 'Header' and 'Body'. Each section has a table with two columns: 'Type' and 'Text / File Name'. Below each table are buttons: 'Add' (with a dropdown arrow), 'Remove', 'Edit', 'Up', 'Down', and 'Convert to ML Prompt'. The 'Add' dropdown menu is open, showing options: 'Add File', 'Add File from Variable', 'Add Text', and 'Add Multilingual Prompt'. At the bottom right are 'Save' and 'Cancel' buttons.

Text tab. Ignore this tab.

The screenshot shows the 'Query Module Properties' dialog box with the 'Text' tab selected. The dialog has tabs for General, Request, Responses, Prompts, Page, and Text. The 'Text' tab contains a checkbox 'Use Visual IVR settings' which is unchecked, and a sub-section 'Text only' which is checked. Below this is a table with two columns: 'Type' and 'Text / File Name'. At the bottom are buttons: 'Add' (with a dropdown arrow), 'Remove', 'Edit', 'Up', 'Down', 'Convert to ML Prompt', 'Save', and 'Cancel'. The 'Add' dropdown menu is open, showing options: 'Add Text' and 'Add Multilingual Prompt'.

Customizing the Script

The generated script contains the input and output parameters that you added in the Request and Response tabs. This section describes how to customize parts of the script. The complete script (Query4.pl) is included in the ZIP file.

Important: Five9 does not correct or accept incorrect specification of the content-type header, such as `content-type: text/xml; charset="utf-8"`. Instead, we require standard use (without using quotation marks). **Example:** `content-type: text/xml; charset=utf-8.`

[Error Codes](#)

[Use Query](#)

[Output Variables](#)

Error Codes. This section describes how errors are indicated. `ERR_CODE__="0"` corresponds to a successful request.

Example

Use case

```
#!/usr/bin/perl
#=====
# Generated by ScriptGen v6.1 (build 0, "OS_BUILD")
# at Mon Jun 17 14:46:32 PDT 2020
#=====
use warnings;
use strict;
use CGI qw(:all);
$|++;
my $cgi = new CGI;
#Variables ERR_CODE__ and ERR_DETAILS are reserved for errors
indication
my $ERR_CODE__="0";
my $ERR_DETAILS__="";
```

Use Query. This section is the only one that you need to modify because the module generates only a commented section. You need to add the code to access and search the database for the variables specified in the comments. In this example, the database is MySQL. The example code that we added is **highlighted**.

```
#=====
=== PLEASE INSERT THE CODE HERE FOR ASSIGNING THE REAL VALUES
=== TO THE FOLLOWING VARIABLES:
=== checking, saving, DB_result
=== SET $ERR_CODE__ = "5" ("User Defined Error")
=== AND $ERR_DETAILS__ TO EXPLANATION STRING IF A WRONG CONDITION
=== SHOULD ARISE. #=== {{{{
=== For more information on using Perl with MySQL, see this website:
=== http://www.sitepoint.com/access-mysql-database-perl/use DBI;

=== Set the connection information and credentials to access the
=== database. In Perl, the datasource name is in this format:
=== $dsn = "dbi:<SQL Platform>:<database_name>:<host_name>:<port>;
=== The default port for MySQL is 3306.
```

```

my $dsn = 'dbi:mysql:example_test:localhost:3306' ;
my $user = 'test_user' ;
my $password = 'test_password' ;
#=== Connect to the database.
my $dbh = DBI->connect($dsn, $user, $password) ;

#=== Prepare and execute the query.
if ($dbh)
{my $sth = $dbh->prepare("SELECT checking_balance, saving_balance FROM
accounts WHERE number=$account_number");
  IF ($sth && $sth->execute ())
  { $DB_result = 1;
    if (!($checking, $saving) = $sth->fetchrow_array()) ($DB_result = 2;
    ($checking, $saving) = (0.00, 0.00));

  }

#=== Disconnect from the database.
  $dbh->disconnect();

}

#=== }}}}}

```

Output Variables. This section contains the checks to verify the data format of the variables (currency and integer) for the parameters to be returned. The print statements describe the format for the parameters, which is an XML structure, as you specified in the Response tab.

```

=====
# Output variable checking.
=====
if (!defined $checking && isvalidcurrency($checking)) {
  $ERR_CODE__ = "3";
  $ERR_DETAILS__ = "Invalid value '$checking' for parameter 'checking' "
;
}
if (!(defined $saving && isvalidcurrency($saving))) {
  $ERR_CODE__ = "3";
  $ERR_DETAILS__ = "Invalid value '$saving' for parameter 'saving'
}
if (!(defined $DB_result && isvalidinteger($DB_result))) {
  $ERR_CODE__ = "3";
  $ERR_DETAILS__ = "Invalid value '$DB_result' for parameter 'DB_result'
}
print qq|Content-Type: text/xml\n\n|;
print qq|<?xml version="1.0"?>;
print qq|<response>;
print qq| <variable>;
print qq| <var name="DB_result" expr = "| . convert2vxml($DB_result) .
qq|>|;
print qq| <varname="checking" expr = "| . convert2vxml($checking) . qq|

```

```

/>|;
print qq| var name="saving" expr = "| . convert2vxml($saving) . qq1"
/>|;
print qq| </variables>|;
print qq| <error code="| . convert2vxml($saving) . qq1" description = "|
. convert2xml($ERR_DETAILS__) . qq1" />|;
print qq|</response>|;

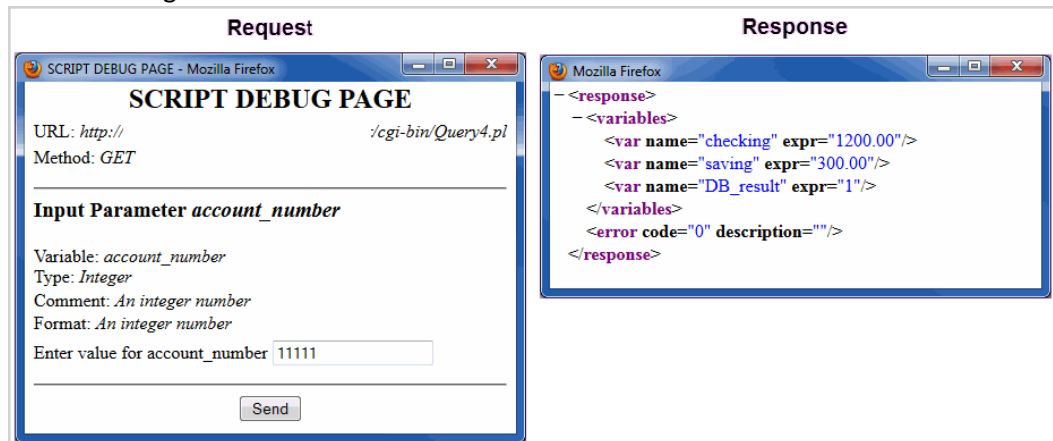
```

Testing Your Implementation

When done with all configuration, test your implementation with valid and invalid input.

Code Script. The HTML file included in the package enables you to debug the code, for example:

- If you send a valid account number, such as 11111, you receive the successful response on the right. In the XML, the `var` elements correspond to the parameters that you entered in the Response tab of the Query module.
- If you send an invalid account number, such as 1111 or 1111a, you receive error messages.



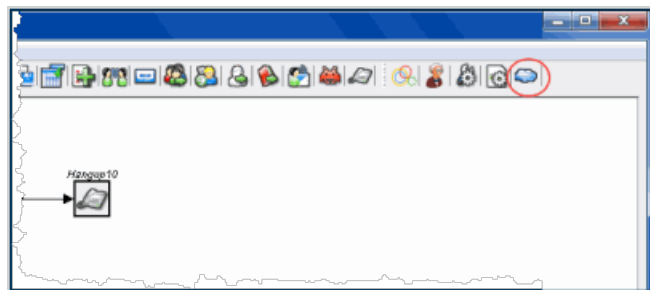
IVR Script. To ensure that the complete flow works as expected, be sure to test your IVR script.

Five9 Salesforce Module

The Five9 Salesforce IVR module enables you to manage interactions and workflow between the IVR and Salesforce. Salesforce must be enabled for your domain to access

the Salesforce module and the associated service scripts. Contact your Five9 representative for additional information.

You can define a Salesforce Object Query Language (SOQL) query, enabling the IVR module to pass the request and return the response with any associated error messaging. The query response is treated as an object that you can route as an interaction. The Five9 Salesforce module is available from the IVR builder toolbar, seen with the Salesforce cloud icon in the figure below.

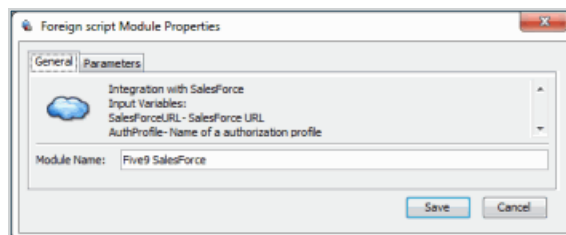


You can drag the Salesforce module to your canvas to build your IVR script, or you can choose one of the predefined **Service Script** examples to use as a template. Right-click or double-click to access the module properties.

You can rename the module in the **General** tab and assign your specific parameters in the **Parameters** tab.

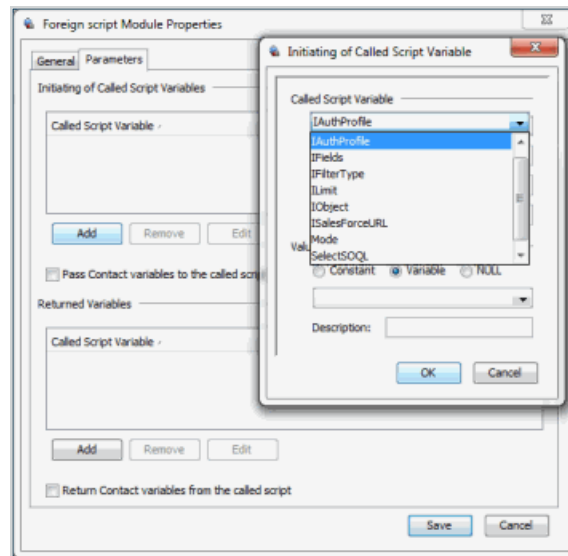
Properties

General Tab



- **Module Name:** Name shown in IVR builder.

Parameters Tab



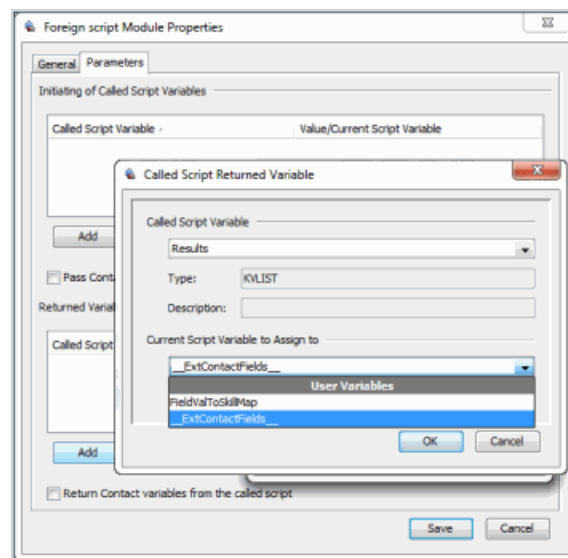
- **Input Parameters:** Contains authorization profile and called script with variable parameters:
 - **IAuthProfile:** Name of an authorization profile configured for Salesforce Gateway in VCC.
 - **IFields:** KVLlist of the object's fields to receive. Keys are the field names. If a value is not empty, it is used for filtering records. All case field's ID included with **Omni.external_id** by default. This field is optional.
 - **IFilterType:** KVLlist of comparison module operation modifiers. This field is optional.
 - **ILimit:** Maximum number of returned records. Default is 1.
 - **IObject:** (Optional) Default is **Case**. Salesforce object name.
 - **ISalesForceURL:** URL of the Salesforce SOQL REST API.
 - **Mode:** **Simple** (default value) or **Advanced**. Advanced mode uses SOQL.
 - **Simple:** Default value. Simple mode uses **IObject**, **IFields**, and **IFilterType**.
 - **Advanced:** Uses a **SelectSOQL** string containing an executable SOQL query.
 - **SelectSOQL:** Executable SOQL for **Advanced** mode.
- Select check box to pass contact variables to the called script.

Example

```
SELECT Id, Name, BillingCity FROM Account LIMIT 3
```

For more information about creating SOQL queries, see [Introduction to SOQL and SOSL](#).

- **Returned Variables:**



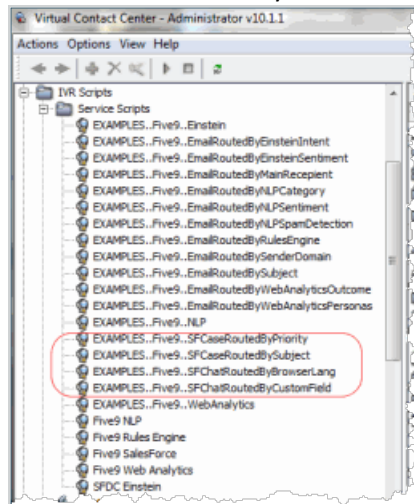
- **Results:** Resulting KVList from query response.
 - `__ExtContactFields__` is the default variable for data exchange with third-party systems.
 - `FieldValToSkillMap` enables additional data mapping.
- **Possible Errors:**
 - **CUSTOM_ERR_1:** Salesforce error.
 - **CUSTOM_ERR_2:** Bad input parameters.
 - **CUSTOM_ERR_3:** Authorization error.

Example IVR Scripts with Salesforce Module

To see a working example before you create your own, navigate to the **IVR > Service Scripts** subfolder to find the following sample scripts:

- SFCaseRoutedByPriority
- SFCaseRoutedBySubject

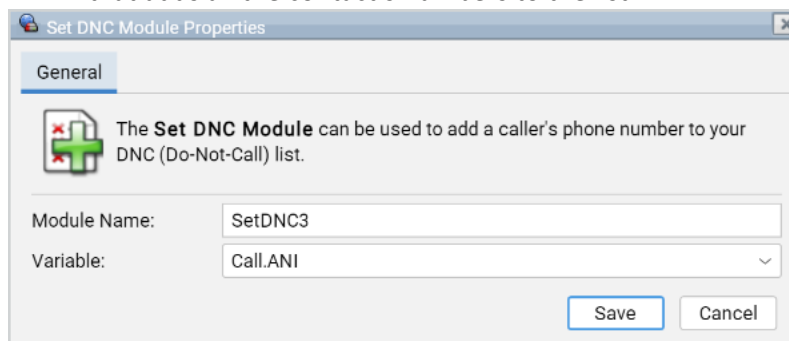
- SFChatRoutedByLang
- SFChatRoutedByCustomField



Set DNC Module

The Set DNC module places a single phone number in your local Do Not Call list. To place the many phone numbers of a single contact in your DNC list, attach the contact record to the call, and select one of these options:

- Use one Set DNC module for each phone number (maximum three modules).
- Create an IVR script that contains post-processing steps, such as a disposition that adds all the contact's numbers to the list.



Use this module as follows:

- If an inbound campaign receives return calls from an outbound campaign, give the caller the option to be placed on your Do Not Call list so that your agents do

not need to spend time to process DNC requests.

- For autodial campaigns, provide to contacts an option to opt out of future campaigns.

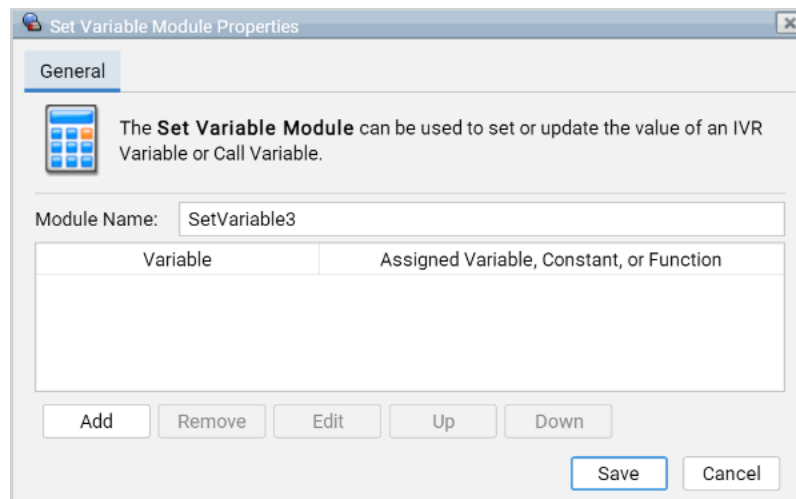
The Variable menu contains a list of all your variables. Select the string variable that contains the phone number to add to your DNC list.

Set Variable Module

The Set Variable module enables you to assign specific values to variables. Typically, use this module to test the value of one variable in the If/Else or Case modules. You assign a value to another variable with the Set Variable module for a subsequent Web request or call routing decision.

Variables are subject to these size limitations: 1M characters for script variables and 2040 characters for call/reporting variables. For an example of the Set Variable module following the If/Else module, see [Simple Campaign Script for Post-Call Surveys](#).

Because you may not use call variables that contain sensitive information in this field, these variables do not appear in the menu even if they exist in the Call Variables folder of the VCC.



The Set Variable Module Properties dialog box is shown. It has a title bar with a close button. The 'General' tab is selected. Below the tab is a description: 'The Set Variable Module can be used to set or update the value of an IVR Variable or Call Variable.' Below this is a text field for 'Module Name' containing 'SetVariable3'. Below the text field is a table with two columns: 'Variable' and 'Assigned Variable, Constant, or Function'. The table is currently empty. Below the table are buttons for 'Add', 'Remove', 'Edit', 'Up', and 'Down'. At the bottom right are 'Save' and 'Cancel' buttons.

Variable	Assigned Variable, Constant, or Function
----------	--

Click **Add**, and select your options.

Function Arguments:		
#1	Constant	q
#2	Constant	w
#3	Constant	e
#4	Constant	r
#5	Constant	

The required data types for your arguments display for each selected function.

- **Variable:** List of your user, contact, IVR and user CAV variables. When you select a variable, the type, such as string, and description are automatically populated.
To ensure that you can save the module variable, follow these rules about sensitive information:
 - If you select a variable that does not contain sensitive information, you may assign only a variable or function value **not** containing sensitive information.
 - If you select a variable that contains sensitive information, you may assign any type of variable or function value.
- **Assigned Value:** Type of value to assign to the variable.
 - **Constant:** Single specified value.
 - **Variable:** List of your contact, system, agent, IVR and user CAV variables. When you select a variable, the description is automatically populated.

- **Function:** List of functions. For more information, see the tables below.
- **NULL**

When using Constant and Function values, you must enter phone numbers according to your domain and VCC configuration:

- If your domain is enabled for the E.164 format, you may need to enter phone numbers in your national format or phone numbers up to 15 digits, starting with the plus (+) sign and the country code. See your administrator.
- If your domain is enabled for the E.164 format, you may need to enter phone numbers in your national format or phone numbers up to 15 digits, starting with the plus (+) sign and the country code. See your administrator.
- If your domain is not enabled for the E.164 format, you must enter 10 digits to conform to the [North American Numbering Plan](#) (NANP).

This table contains the possible operations for each data type.

Left Variable Type	Function	Description	Argument Types	Exceptions
INTEGER	DIFFERENCE	Number of days between two dates.	DATE, DATE	Arg2>Arg1, Arg1 or Arg2 is NULL
	DIFFERENCE	Number of seconds between two times.	TIME, TIME	Arg2>Arg1, Arg1 or Arg2 is NULL
	DIFFERENCE	Subtracts second argument from first one.	INTEGER, INTEGER	Arg2>Arg1, Arg1 or Arg2 is NULL
	LEN	Returns the number of characters of the specified string expression, excluding trailing blanks.	STRING_EXPR	Arg1 is NULL
CURRENCY	SUM DIFFERENCE	Adds or subtracts arguments.	CURRENCY, CURRENCY	Arg1 or Arg2 is NULL
	PRODUCT	Multiplies the currency by an integer.	CURRENCY, INTEGER	Arg1 or Arg2 is NULL

Left Variable Type	Function	Description	Argument Types	Exceptions
	QUOTIENT	Divides the currency by an integer.	CURRENCY, INTEGER	Arg1 or Arg2 is NULL
DATE	ADD	Adds Arg2 days to Arg1 date.	DATE, INTEGER	Arg1 or Arg2 is NULL
INTEGER	LOCATE	Returns the position of the first occurrence of <i>substring</i> within <i>string</i> .	SUBSTRING, STRING	Arg1 or Arg2 is NULL
	PRODUCT	Multiplies arguments.	INTEGER, INTEGER	Arg1 or Arg2 is NULL
	QUOTIENT	Divides first argument by second one.	INTEGER, INTEGER	Arg2=0, Arg1 or Arg2 is NULL
	SUM	Adds arguments.	INTEGER, INTEGER	Arg1 or Arg2 is NULL
	TOINTEGER (CURRENCY(\$))	Converts value into USD currency integer.	STRING_EXPR	Arg1 is NULL
	TOINTEGER (CURRENCY(£))	Converts value into GBP currency integer.	STRING_EXPR	Arg1 is NULL
	TOINTEGER (CURRENCY(€))	Converts value into Euro currency integer	STRING_EXPR	Arg1 is NULL
	TOINTEGER (DATE)	Converts date to integer	DATE_EXPR	Arg1 is NULL
	TOINTEGER (NUMBER)	Converts number to integer. Use this to convert floating-point values, such as 31.2, to an integer.	NUMBER_EXPR	Arg1 is NULL
	TOINTEGER (STRING)	Converts string expression to integer. Use this to convert whole numbers, such as 31, to integer.	STRING_EXPR	Arg1 is NULL
	TOINTEGER (TIME)	Converts time expression into integer.	TIME_EXPR	Arg1 is NULL
NUMBER	SUM DIFFERENCE PRODUCT QUOTIENT	Adds, subtracts, multiplies, or divides arguments.	NUMBER, NUMBER	Arg1 or Arg2 is NULL

Left Variable Type	Function	Description	Argument Types	Exceptions
	LEN	Returns the number of characters of the specified string expression, excluding trailing blanks.	STRING_EXPR	Arg1 is NULL
	RANDOM	Returns numeric value between 0 (inclusive) and 1 (inclusive). The result can be assigned to Numeric IVR variables or NUMBER call variables.	None	
STRING	BASE64DECODE	Decodes STRING IVR or call variable.	STRING	STRING is NULL
	BASE64ENCODE	Encodes STRING IVR or call variable.	STRING	STRING is NULL
	CONCAT	Concatenates up to five strings.	STRING, STRING, STRING, STRING, STRING	Any of Arg1 through Arg5 are defined by Variable having a NULL value
	GET	Returns the value to which the specified key is mapped, or null if this kv-pair contains no mapping for the key.	KVLIST, STRING	Arg1 or Arg2 is NULL
	GET_KEY	Returns nth key of kv if $0 \leq \text{nth} < \text{SIZE}(\text{kv})$. Activates exceptional exit otherwise.	KVLIST, INTEGER	Arg1 or Arg2 is NULL
	HASHBYTES	Hash the string using algorithm method: MD2 MD4 MD5 SHA SHA1 SHA_256 SHA_512	<algorithm>, STRING	Arg1 is NULL or incorrect method
	LEFT	Returns the leftmost characters of a string.	STRING_EXPR, LENGTH	Arg1 is not a STRING IVR or call variable Arg2 is negative
	PAD	Pads a string to a new length extending the string to the right.	STR, INT (Length), STR	Arg1 or Arg2 is NULL

Left Variable Type	Function	Description	Argument Types	Exceptions
		Example: PAD("1234",10,"X") results in: "1234XXXXXX"	(Padstring)	
	PUT	Associates the specified value with the specified key. Returns an old value if the key existed before the function's invocation, or NULL otherwise.	KVLIST, STRING (key), STRING (value)	Arg1 is not in KVLIST Arg2 is invalid Any Argument is NULL
	RIGHT	Returns the rightmost characters of a string.	STRING_EXPR, LENGTH	Arg1 is not a STRING IVR or call variable Arg2 is negative
	SUBSTRING	Returns string value which can be assigned to a STRING IVR variable or a STRING call variable expression.	EXPRESSION, START, LENGTH Indexed starting at zero (0), from left.	Arg1 is not a STRING IVR or call variable Arg2 is greater than number of characters Arg3 is not a positive or whole number
	TO_E164PHONES	Returns country code of phone number in E164 domains.	STRING, STRING	Arg1 or Arg2 in NULL or CAV is UNDEFINED
	TOSTRING (CURRENCY(\$))	Converts value to USD currency string.	STRING	Arg1in NULL
	TOSTRING (CURRENCY(£))	Converts value to USD currency string.	STRING	Arg1in NULL
	TOSTRING (CURRENCY(€))	Converts value to USD currency string.	STRING	Arg1in NULL
	TOSTRING (DATE)	Converts date value to string.	DATE	Arg1in NULL
	TOSTRING (DATE, STRING)	Converts date value to string.	DATE, STRING	Arg1 or Arg2 is NULL
	TOSTRING (INTEGER)	Converts INTEGER value to string.	INTEGER	Arg1 is NULL

Left Variable Type	Function	Description	Argument Types	Exceptions
	TOSTRING (KVLIST)	Converts KVLIST to string.	KVLIST	Arg1 is NULL
	TOSTRING (NUMERIC)	Converts numeric value to string value.	NUMERIC	Arg1 is NULL
	TOSTRING(TIME)	Converts TIME value to string value.	TIME	Arg1 is NULL
	TOSTRING(TIME, STRING)	Converts TIME value to string value.	TIME, STRING	Arg1 or Arg2 is NULL
	TO_UTC_TIME (TIME, STRING)	Converts TIME to local time; that is, the GMT setting for your Time Zone in the IVR schedule of the campaign properties.	TIME, STRING	Arg1 or Arg2 is NULL
TIME	ADD	Adds Arg2 seconds to Arg1 time.	TIME, INTEGER	Arg1 or Arg2 is NULL

Converting Integers

To change a data type, use the data transformation methods in this table. Numbers with decimals are not converted to integers.

Variable Type	Function	Arguments	Comment	Exceptions
INTEGER	TOINTEGER	NUMBER	Fraction truncated.	Arg is NULL
	TOINTEGER	CURRENCY	Fraction truncated.	Arg is NULL
	TOINTEGER	STRING	Until first non-digit; can be empty.	Arg is NULL or contains non-digit characters.
	TOINTEGER	DATE, STRING	Second argument: d, such as 31 m, such as 12 y, such as 2007	Arg1 or Arg2 is NULL Wrong Arg2
	TOINTEGER	TIME, STRING	Second argument: h, such as 18 m, such as 45	Arg1 or Arg2 is NULL. Wrong Arg2
NUMBER	TONUMBER	INTEGER		Arg is NULL
	TONUMBER	CURRENCY		Arg is NULL
	TONUMBER	STRING	Period (.) for decimals.	Arg is NULL
CURRENCY	TOCURRENCY	INTEGER	.00 for fractions.	Arg is NULL
	TOCURRENCY	NUMBER	Fraction truncated to 2 digits.	Arg is NULL
	TOCURRENCY	STRING	Period (.) for decimals.	Arg is NULL

Variable Type	Function	Arguments	Comment	Exceptions
STRING	TOSTRING	INTEGER		Arg is NULL
	TOSTRING	NUMBER		Arg is NULL
	TOSTRING	CURRENCY		Arg is NULL
	TOSTRING	DATE	yyyy-mm-dd = ymd	Arg is NULL
	TOSTRING	DATE, STRING	Second argument: d, such as Monday dmy, such as 31/12/2007 m, such as June md, such as June, 14 mdy, such as 12/31/2007 ymd, such as 2007-12-31	Arg1 or Arg2 is NULL wrong Arg2
	TOSTRING	TIME	hh:mm = hm24	Arg is NULL
	TOSTRING	TIME, STRING	Second argument: hm12, such as 6:45 PM hm24, such as 18:45	Arg1 or Arg2 is NULL. wrong Arg2
DATE	TODATE	STRING		Arg is NULL,
TIME	TOTIME			unsupported transformation
WAV Files	WAV2BASE64 WAV2BASE64	STRING	Encodes in ASCII format.	Arg is NULL, unsupported transformation

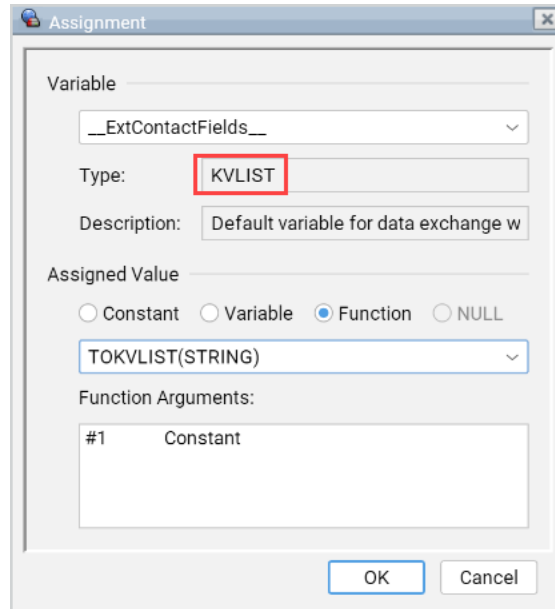
Using the KVLIST Data Type Functions

To pass sets of data fields to and from IVR, use collections of values with functions for accessing, adding, and deleting elements with keys. You can iterate all elements in the collection and convert the collections to strings to send/receive them to/from external systems. To do so, use the KVLIST data type, which uses collections of string key-value pairs instead of elementary data.

[Defining a KVLIST Collection](#)
[Using KVLIST Variable Functions](#)

Defining a KVLIST Collection

You can introduce a KVLIST collection by defining a variable with KVLIST data type selected. For information on adding or creating variables, see [Adding Variables](#).



The screenshot shows the 'Assignment' dialog box with the following configuration:

- Variable:** A dropdown menu showing `__ExtContactFields__`.
- Type:** A dropdown menu showing `KVLIST`, which is highlighted with a red rectangle.
- Description:** A text field containing `Default variable for data exchange w`.
- Assigned Value:** A section with four radio buttons: `Constant`, `Variable`, `Function` (selected), and `NULL`.
- Function Arguments:** A dropdown menu showing `TOKVLIST(String)`.
- Function Arguments:** A table with one row: `#1` and `Constant`.
- Buttons:** `OK` and `Cancel` buttons at the bottom right.

These variables may have an initial value defined as a JSON-formatted string with a collection of key-value pairs, where both keys and values are strings.

Example

A variable named `Person` contains key-value pairs that describe a person:

```
Person = {"last_name": "Jones", "first_name": "Betty", "age":  
"31", "phone": "650.321.4800"}  
"last_name", "first_name", "age", and "phone" are keys for values "Jones",  
"Betty", "31", and "650.321.4800".
```

KVLIST variables without a default value are considered empty variables.

Assignment `{}` is an empty KVLIST but a non-empty variable.

Using KVLIST Variable Functions

Specific functions are used in conjunction with the KVLIST data type to return a JSON string of key-value pairs enabling you to store multiple data elements in a single IVR

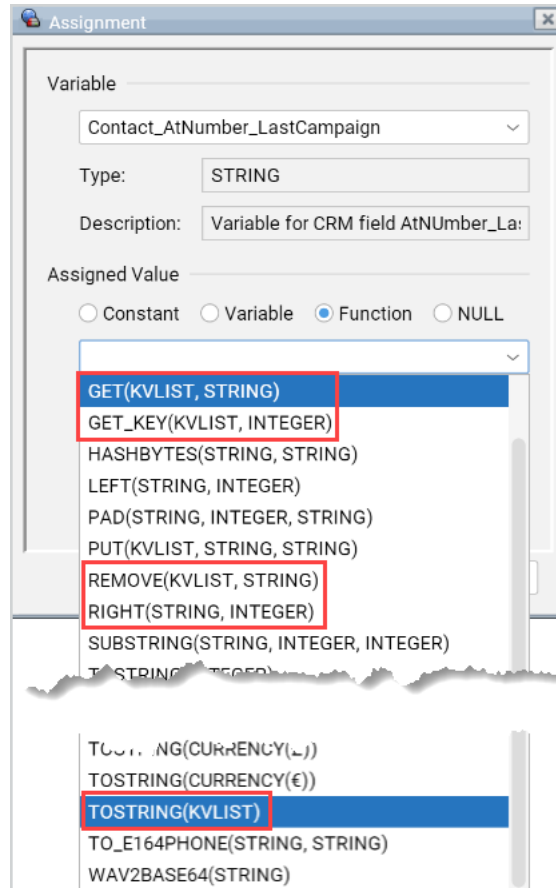
variable. Benefits of using the KVLIST data type include:

IVR can dynamically expand or contract the volume of variables, or data elements, used rather than manually storing every possible variable value.

JSON response to IVR Query module requests can be more easily parsed than standard elementary pairing with string pattern matches.

Specific functions (see function descriptions below) with a KVLIST data type return a JSON string of key-value pairs, enabling you to store multiple data elements in a single IVR variable:

- To GET the *value* of a key.
- To GET a *key* according to index, or position.
- To PUT, or write, a new key-value pair.
- To REMOVE a KV pair from a KVLIST.
- To TRANSFORM a KVLIST to a regular string.
- To return the SIZE of a KVLIST as an integer.
- To TRANSFORM a string to a KVLIST.



You can use the KVLIST of variables only with these functions:

Returns	Function	Description
STRING	GET (KVLIST kv, STRING key)	Returns the value for mapping the specified key or NULL if this key-value pair contains no mapping for the key.
	GET_KEY (KVLIST kv, INTEGER nth)	Returns nth key of kv if $0 \leq \text{nth} < \text{SIZE}(\text{kv})$. Activates an exception exit otherwise.
	PUT (KVLIST kv, STRING key, STRING)	Associates the specified value with the specified key. Returns an old value if the key existed before the function's invocation or NULL.

Returns	Function	Description
	value)	
	REMOVE (KVLIST kv, STRING key)	Removes the mapping for a key from kv if it is present. Returns removed value or NULL if not present.
	TO_STRING (KVLIST kv)	Returns JSON representation of kv. Activates an exception exit if kv is NULL.
INTEGER	SIZE (KVLIST kv)	Returns the number of key-value mappings in kv.
KVLIST	TO_KVLIST (STRING arg)	Parses arg to receive KVLIST. Activates an exception exit if arg is in the incorrect format.

Skill Transfer Module

Use the Skill Transfer module to transfer calls to skill groups and configure the queue according to agent availability. When you add multiple skills to this module, any agent with one of the skills can receive calls from the module. After the call transfers to an agent, the IVR script is complete.

[General Tab](#)
[Skills Tab](#)
[Announcements Tab](#)
[Estimated Wait Time](#)
[Priority Tab](#)
[Connectors Tab](#)
[Queue Callback Tab](#)
[Dispositions Tab](#)
[Disposition Flow](#)

Skill Transfer Module Properties

General Skills Announcements Priority Connectors Queue Callback Disposition

 The **Skill Transfer Module** can be used to queue callers and play in-queue hold music and announcements until the callers are routed to an available appropriately-skilled agent.

Module Name: SkillTransfer2

Call Queuing

Max Queue Time (mmm:ss)

If there are available agents in queue 010:00

☒ Queue calls when agents in queue, but unavailable (on call) 010:00

☐ Queue calls when no agents in queue (not ready / logged out) 010:00

Termination Digit: N/A

Queue Callback

☒ On queue timeout expiration ☒ On digit receiving: #

☒ Record caller name for agent assigned to QCB

Transfer to Agent

Distribution Algorithm: Longest Wait – All Calls

Time Frame: 15 Minutes

Max Ring Time (secs): 15 Pause Before Transfer (secs): 0

☒ Set agent to Not Ready if no answer

Transfer to Voicemail

☒ On queue timeout expiration ☐ On digit receiving:

Voicemail Box: ☐ Skill ☒ Personal ☒ Variable: CavGroup1.DateTime

Action for Recorded Files: Keep as Recording

☒ Clear User Input

Save Cancel

General Tab

- **Max Queue Time:** Maximum time allowed for calls in a queue. The default is 10 minutes for each setting. The maximum is 999 minutes and 99 seconds.
 - If there are available agents in queue: Agents are logged in but do not answer the call or the station is not connected.
 - Queue calls when agents in queue but unavailable (on call): Agents are currently logged in but busy with other calls. If you enable this option, the minimum value is 00:01 second.
 - Queue calls when no agents in queue (not ready / logged out): Agents are logged in but not ready, or all agents are logged out. If you enable this option, the minimum value is 00:01 second.

Afterward, depending on the settings below, calls are transferred to an agent or to voicemail according to the IVR flow.

- **Termination Digit:** Phone key that enables callers to exit the queue. Afterward, depending on the settings below, calls are transferred to an agent or to a voicemail module.
 - Enabled: #, *, and digits from 1 through 9.
 - Disabled: N/A.
- **Queue Callback:** Phone key that enables callers to receive a callback. Callers retain their position in the queue without staying on hold. The Queue Callback feature is disabled when both Queue Callback options are disabled.
 - **On queue timeout expiration:**
 - If the caller chose a queue callback, the call remains in the queue and is processed as queue callback.
 - If the caller declined the queue callback, the call either returns to IVR or is transferred to voicemail.
 - **On digit receiving:** Number callers press to receive a callback. If enabled: #, *, and digits from 1 through 9.
To prevent errors, the user interface does not display the Termination Digit in the Queue Callback Digit menus.
 - **Record caller name for agent assigned to QCB:** Caller can record a name that is forwarded with the queue callback request. Before agents return the call, the recording is played for agents when Automatic Answer is on. This enables agents to announce the name of the person who left the queue callback to the person who answers the call. Agents can also replay the recording to the person who answers the phone if clarification is

needed.

Contact your Five9 account representative for additional information.

- **Distribution Algorithm:** Method used by the ACD to select the agent who receives the call:
 - **Longest Wait - All Calls:** Agent with the longest idle time since the last call.
 - **Longest Wait - Non-Manual:** Agent with the longest idle time, excluding manual calls, such as callbacks.
 - **Round Robin:** Agent with the longest idle time among those logged into the queue. When agents log into the queue, they have the shortest idle time. The first queued call goes to the agent with the highest idle time. This agent is removed from the list. The process continues from the highest to the shortest idle time.
 - **Min Calls Handled:** Agent who handled fewer queued calls than other agents during the selected interval.
 - **Min Handle Time:** Agent with the lowest total call handle time in the queue during the selected interval.
- **Time Frame:** Interval (15 minutes to 24 hours) for the Distribution Algorithm option.

Note: We recommend setting this interval to 24 hours.

- **Max Ring Time:** Maximum number of seconds allowed for the agent to answer a call.
- **Pause Before Transfer:** Number of seconds before transferring a call to an agent.
- **Set Agent to Not Ready if No Answer:** Whether to change the status of the agent to Not Ready if the agent doesn't answer a call within the maximum ring time allowed.
- **On queue timeout expiration:** Whether to transfer calls to voicemail when the maximum queue time has expired. This option activates the type of voice mailbox: personal or skill.
- **On digit receiving:** Number callers press to activate the transfer to the voicemail box: #, *, or digits from 0 through 9. This option activates the type of voicemail box: personal or skill.

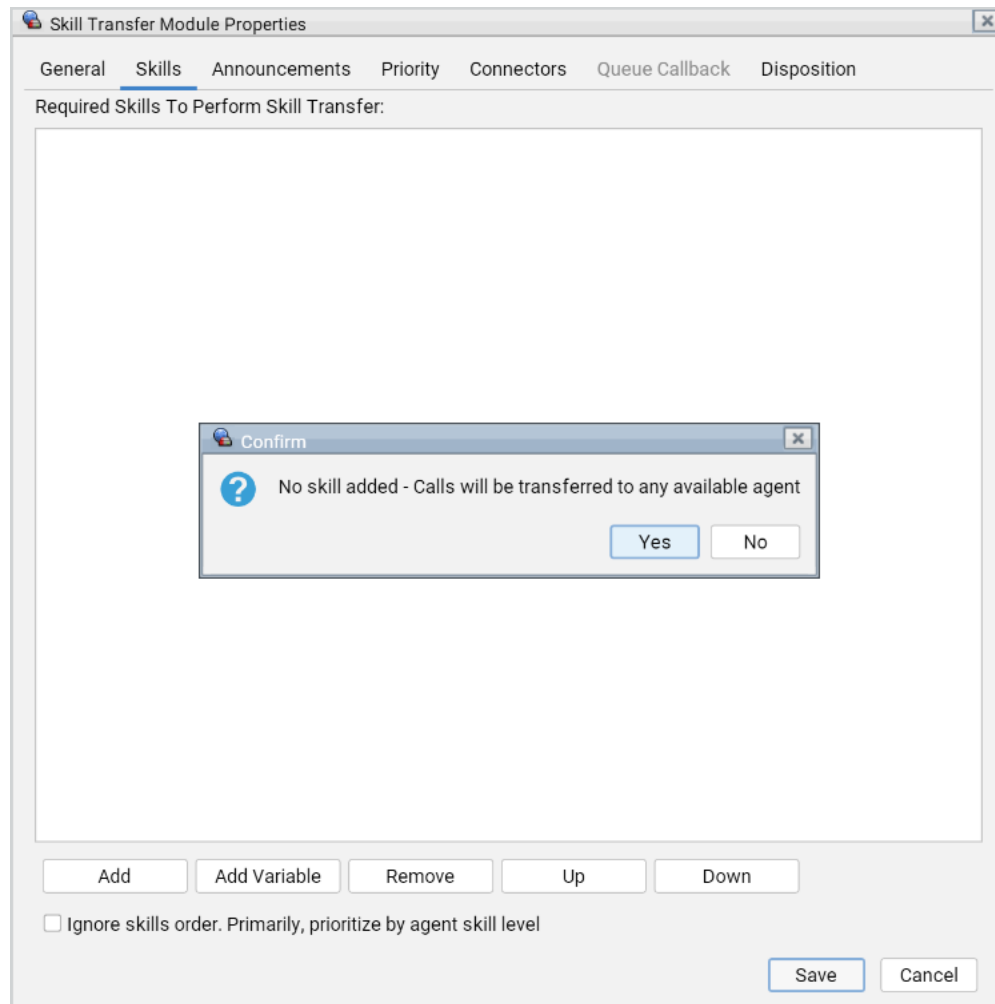
- **Voicemail Box:** Type and voice mailbox to activate for the transfer:
 - **Skill:** list of all skills in the domain.
 - **Personal:** list of all users who have permission to receive transfers.
 - **Variable:** list of defined variables. Select this option to obtain the value for Skill or Personal from the selected variable.
- **Action for Recorded Files:** Method for handling the recorded files gathered by the previous modules. See also [Using Recorded Files](#).
 - **Keep as Recording:** Save recorded files as a recording to access from reporting. The recorded files are not sent to the agents or skills.
 - **Send to Agent:** Routes recorded files to the agent. If agents are not available and the voicemail option is enabled, the recorded files become part of the voicemail message.
- **Clear User Input:** Whether information entered by the caller before reaching this module is erased.
 - **Enabled:** Delete all user input before this module.
 - **Disabled:** Do not enable if you have a prompt that provides instructions, such as pressing a key (termination digit) to exit the queue, before this module.

Skills Tab

List of queues and variables for call transfers. To obtain useful reports, Five9 recommends that you add only one queue in each module. You can prioritize the list as needed.

- Select one or more queues by name.
- Select one or more variables that contain the name of a queue as a string. If a variable contains the name of an invalid queue, the call will not transfer.

If you don't select queues, calls route to users not assigned to a queue.



The unassigned queue is labeled Default in the softphone. This feature enables you to use a default queue to send calls to agents and supervisors not assigned queues or when agents assigned a particular queue are not available.

Announcements Tab

Recorded set of greetings or custom prompts played until one of these events occur:

- Agent answers.
- Call queue timeout expires.
- Customer enters termination digit.
- Call transfers to voicemail.

You can enable up to fifteen recordings.

The screenshot shows the 'Skill Transfer Module Properties' dialog box with the 'Announcements' tab selected. The 'General' tab is also visible. The 'Announcements' tab contains a checkbox for 'Enable music on hold' which is checked. Below this is a dropdown menu for 'Estimated Wait Time Announcement' set to '[default]' and a 'New...' button. The 'In Queue Announcements' section contains a table with columns: 'Enabled', 'Loop', 'Timeout (secs)', and 'Voice File to Play'. The table has 15 rows, each with an unchecked 'Enabled' checkbox, an unchecked 'Loop' checkbox, a '1' in the 'Timeout (secs)' column, and 'Estimated Wait Time' in the 'Voice File to Play' column. At the bottom right of the dialog are 'Save' and 'Cancel' buttons.

Enabled	Loop	Timeout (secs)	Voice File to Play
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time
☐	☐	1	Estimated Wait Time

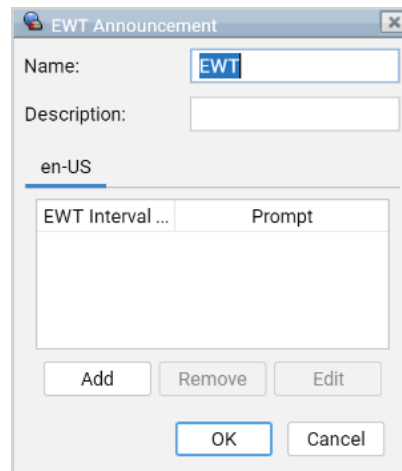
Create an announcement to play a prompt or precede this module with a Play module that contains a prompt.

- **Enable music on hold:** Whether to enable music for callers on hold. If enabled, the default music is played between each recording.
- **Estimated Wait Time Announcement:** Message that indicates to the customer the possible waiting time (EWT) until an agent answers the transferred call. The announcement is appended to the prompt that you select.

Each announcement is actually a set of announcements

The menu contains a default announcement or a list of announcements that you create for different durations or events.

- 1 To customize your list of EWT prompts, click **New > Add**.

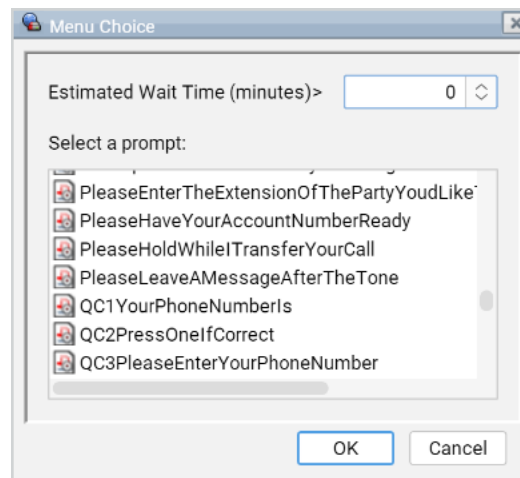


The screenshot shows the 'EWT Announcement' dialog box. It has a title bar with a close button. Inside, there are two text input fields: 'Name:' with 'EWT' entered and 'Description:' which is empty. Below these is a language dropdown menu set to 'en-US'. Underneath is a table with two columns: 'EWT Interval ...' and 'Prompt'. The table is currently empty. At the bottom of the dialog are three buttons: 'Add', 'Remove', and 'Edit'. At the very bottom are 'OK' and 'Cancel' buttons.

- 2 Name and describe the announcement.

The name will be displayed in the Estimated Wait Time Announcement field of the Announcements tab.

- 3 Click **Add**.



The screenshot shows the 'Menu Choice' dialog box. It has a title bar with a close button. Inside, there is a dropdown menu for 'Estimated Wait Time (minutes)>' set to '0'. Below this is a section titled 'Select a prompt:' followed by a list box containing several prompts, each with a small icon to its left: 'PleaseEnterTheExtensionOfThePartyYoudLike', 'PleaseHaveYourAccountNumberReady', 'PleaseHoldWhileITransferYourCall', 'PleaseLeaveAMessageAfterTheTone', 'QC1YourPhoneNumberIs', 'QC2PressOnelfCorrect', and 'QC3PleaseEnterYourPhoneNumber'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

- 4 Select a prompt and an estimated time.
- 5 Click **OK** twice.

- **In Queue Announcements:**
 - **Enabled:** Whether to enable the recording. When a recording is enabled, the menus for the recording are activated.

- **Loop:** Whether to play the recording multiple times.
- **Timeout:** Interval from 0 to 3600 seconds between the end of the previous recording and the beginning of the current recording. If the time-out is 0, the current recording is played without pause after the previous recording. If the Loop option is enabled, the time-out also determines the time-out from the end of the previous loop iteration. The default time-out is 1.

The time-out for the first enabled recording is calculated from the moment that the call enters this module.

- **Voice File to Play:** Possible recording options to play:
 - Estimated Wait Time (default prompt)
 - System prompt
 - Custom prompt in your Prompts folder

Estimated Wait Time

Estimated Wait Time (EWT) uses a sophisticated algorithm to predict the wait time for callers within the skill and uses these two consistent metrics:

- EWT = The position of the caller in the queue divided by the Skill Answer Rate. EWT data is pulled from the last 15-minute interval. If the calculated EWT value is less than 60 seconds, the final value is always 60 seconds (this approach applies whether one or more skills are included in the Skill Transfer module). This default calculation applies to queues with >30 calls in the last 15 minutes.

More granular logic is used for <15 or 15–30 calls entering the queue in the last 15 minutes.

- Calculations for **<15 calls in the last 15 minutes:**
 - If calls are immediately assigned to an agent: EWT = 60 seconds
 - If the call is queued and there are no available agents: EWT = current Answer Rate (no less than 60 seconds) or 10 minutes (default) if no calls were in the queue in the last 15 minutes
 - If call is queued and agents are available: EWT = current Answer Rate minus the caller's elapsed Wait Time or 5 minutes minus the caller's elapsed Wait Time if no calls were in the queue in the last 15 minutes
- Calculations for **15–30 calls entering the queue in the last 15 minutes:** the default EWT calculation minus the number of seconds since the last call entered the queue

- **Skill Answer Rate:** The average number of calls answered per minute, using the statistical data collected in the last 15-minute period.

Example

During the last 15-minute period, there were 45 calls answered for a specific skill. The Skill Answer Rate for that skill is 3 per minute: $45 \div 15 = 3$

For the 6th caller waiting in queue, EWT is 2 minutes: $6 \div 3 = 2$

For the 9th caller waiting in queue, EWT is 3 minutes: $9 \div 3 = 3$

Priority Tab

Amount, from 1 (lowest) to 100 (highest), that the call's base priority is increased or decreased. You can use a specific number, or a contact, system, or call variable. Calls with higher priority are answered first.

The base priority is either 10, the default, or the value that is set as Initial Call Priority in the associated campaign profile. See the [Campaign Administrator's Guide](#) for more information. The resulting value must still be within the range one to 100.

If your script has multiple IVR modules that route the calls to the same queue, the behavior is as in the following example.

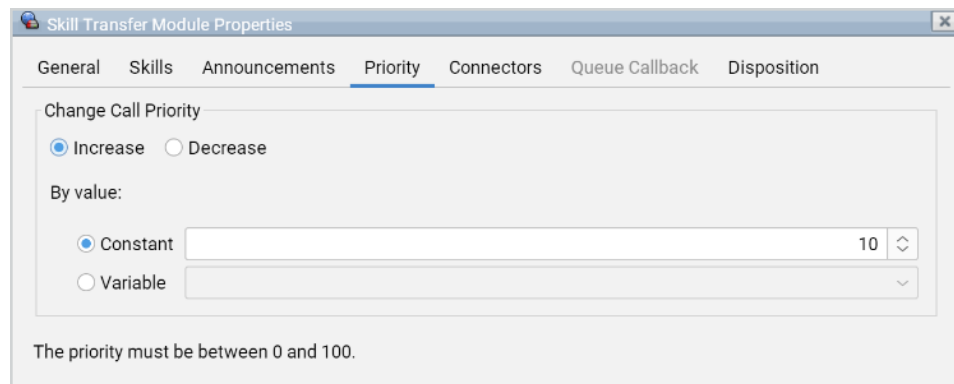
- If both modules have the default priority settings, calls are prioritized by the wait time in the skill transfer module that they are in. So calls that are already waiting in SkillTransferA are routed to Skill W before a call that is just entering SkillTransferB.
- If SkillTransferB has a higher Priority setting, all calls in SkillTransferB are routed to Skill W before any calls from SkillTransfer A are sent there. Example You have SkillTransferA and SkillTransferB, both routing calls to Skill W.

Note

The Skill Transfer module has a default Priority setting of **Increase By: 10**. So if the campaign has no associated campaign profile and you do not alter the module's default setting, the call has a priority of 20.

Example

You have multiple IVR branches for the same agents. You want calls from one branch to be answered first. In this case, assign a higher priority to that branch.



The screenshot shows the 'Skill Transfer Module Properties' dialog box with the 'Priority' tab selected. The 'Change Call Priority' section has 'Increase' selected with a radio button. Below it, 'By value:' is set to 'Constant' with a text box containing '10'. The 'Variable' option is also available. A note at the bottom states: 'The priority must be between 0 and 100.'

Note

The priority of text items routed to agents who can manually select text interactions is the same as that for voice interactions. All interactions are created with default priority 50, which can be changed by Engagement Workflow. Filtering works as expected when items in the view contain interactions that were processed with Engagement Workflow and others that did not.

Connectors Tab

List of connectors that can forward call-related data for reports. For example, a connector can send the call duration or queue time to an external URL when the call is assigned a disposition, such as when the caller abandons the queue.

Queue Callback Tab

This tab is available when you select the Queue Callback in the General tab.

If agents are not available when customers call your company, your customers are placed in a queue. They can continue to wait in the queue, leave a voicemail message, or request a callback. You can dynamically display an estimated waiting time to customers by using Visual IVR for chat or video chat interactions.

The process starts when callers press the queue callback digit that you selected to request a callback instead of remaining on hold until an agent becomes available. For an example flow, see the Queue Callback Flow shown below.

For an example IVR, see [Creating an IVR Script with Queue Callback](#).

The screenshot shows the 'Skill Transfer Module Properties' dialog box with the 'Queue Callback' tab selected. The dialog has several sections: 'General' (with 'Initial Callback Number from' set to 'Call.number' and 'Store Callback Number in' as an empty dropdown), 'Prompts' (with five rows for 'Playing the Phone Number', 'Confirming the Phone Number', 'Entering the Phone Number', 'Recording Caller's Name', and 'Confirmation and Thank You', each with a text field and a 'Select' button), 'Timeouts' (with 'Disable Queue Callback after' set to 0 days, 5 hours, 0 minutes, and 0 seconds, and 'After Digit Timeout (secs)' set to 5), and 'TCPA Consent' (with a checkbox for 'TCPA Consent' and a text area for 'TCPA Consent Text' containing a sample consent message). 'Save' and 'Cancel' buttons are at the bottom right.

- **Initial Callback Number from:** Variable that contains the caller's initial phone number.

Because you may not use call variables that contain sensitive information in this field, these variables do not appear in the menu even if they exist in the Call Variables folder of the VCC.

- **Store Callback Number in:** Variable that contains the phone number specified by the caller for the callback.

Important

When caller enters their callback number, this entry must be followed by a hash symbol termination entry (#).

- **Allow collection and callback of international numbers:** Whether to store phone numbers from outside your country so that you may return those calls. If you do not enable this feature, you can return only national calls. To use this feature, you must have enabled the campaign for callbacks. For more information, see *Creating IVR Schedule Rules* in the [Campaign Administrator's Guide](#).
- **Prompts:** Prompt that you select for each part of the process. Each menu contains the list of prompts in your Prompts folder:
 - Playing the Phone Number
 - Confirming the Phone Number
 - Entering the Phone Number
 - Confirmation and Thank You
- **Timeouts:** Expiration time for the queue callback request.
 - **Disable Queue Callback after:** Expiration day and time.
 - **After digit timeout:** Number of seconds between digits when DTMF entry is used.
- **TCPA Consent:** Whether to enable the feature. This feature applies to Visual IVR only. This option enables the text field.

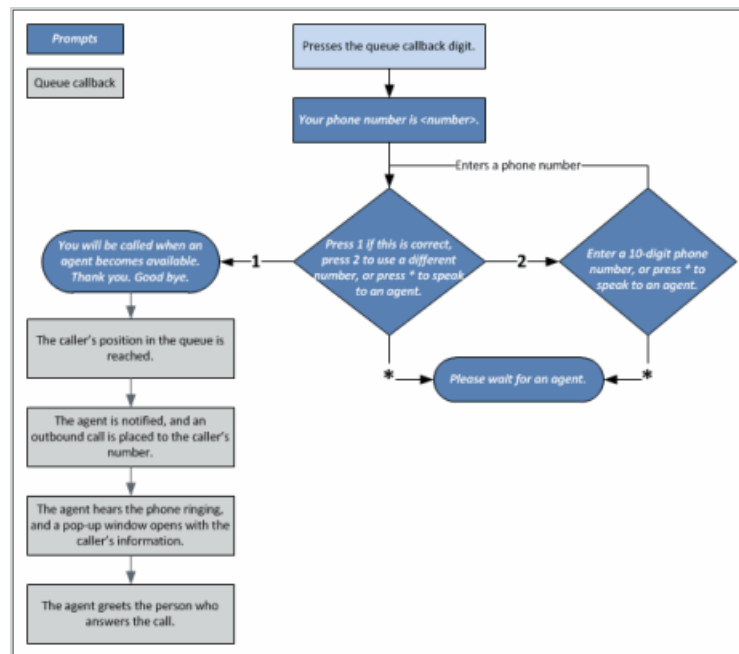
TCPA information is stored in the call log with the date, the name of the contact, and the text of the consent.
- **TCPA Consent Text:** Telephone Consumer Protection Act (TCPA) text that you may modify as needed to comply with TCPA regulations.

Dispositions Tab

The default disposition is Abandon. Be sure to create disposition codes specific to callbacks so that your agents can finish processing the call appropriately. For more information, see [Selecting Dispositions](#).

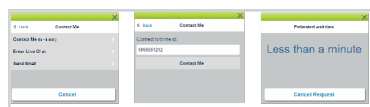
Disposition Flow

This figure shows an example queue callback flow that occurs when a customer calls, but all agents are busy. The diagram starts when the customer enters the queue callback digit.



The next available agent receives a callback request that contains the information about the caller and the call that generated the request.

This shows an example queue callback that appears as a sequence of 3 windows in a Visual IVR:



- First window: Available types of communication: queue callback, chat, and email.
- Second window: Queue callback number or chat window, for example, and Back and Forward options. The number1 contact field is displayed if it is available.
- Third window: Estimated waiting time for a queue callback or chat. The time is updated every 15 seconds until the callback occurs.

System Info Module

The System Info module enables you to retrieve the properties and statistics of users, skills, voicemail messages, EWT, skill groups, and campaigns. You may use the returned variables in subsequent IVR modules.

Example

You can use this module to determine the number of agents logged in to a particular queue. For a queue that is normally staffed, the absence of logged-in agents may indicate a network failure or an emergency evacuation of your contact center. In this instance, you can play an emergency prompt to your callers to explain the absence of agents. See [Defining Search Conditions](#) for configuration details.

When using Visual IVR scripts, you can add images to pages and menus to enhance your brand or to describe a product or service; however, you cannot add links to images. For more information, see [Page Tab](#).

Note: You cannot delete a System Info module if the script contains an Update System Info module with a reference to the System Info module.

The screenshot shows the 'System Info Module Properties' dialog box with the 'General' tab selected. The dialog contains the following fields and options:

- Module Name:** SystemInfo3
- System Object:** Users (dropdown menu)
- Lookup Condition:** (empty text field with a search icon button)
- Returned Parameters:** A table with two columns: 'Parameter' and 'Variable'. Below the table are 'Add', 'Remove', and 'Edit' buttons.
- If Multiple Records Match the Condition:**
 - ☒ Raise an Exception
 - ☐ Save Number of Records to: (empty text field)
- Buttons:** 'Save' and 'Cancel' at the bottom right.

Properties

[General Tab](#)

[Prompts Tab](#)

[Page Tab](#)

General Tab

- **Module Name:** Name that represents the role of the module in the IVR script. In Visual IVR scripts, the name is converted to text that appears in the heading of the IVR window.
- **System Object:** Available system objects:
 - Users
 - Skills
 - Voicemails
 - Statistics: EWT (see [Estimated Wait Time](#))
 - Statistics: Skills
 - Statistics: Campaigns
- **Lookup Condition:** Creates a search condition. If you do not specify a condition, all objects return. This condition can contain user-defined filter criteria, logical operations, and a sort order. For more information, see [Defining Search Conditions](#).
- **Returned Parameters:** Parameters and variables that contain the results. When an object is found, the module assigns the information to a variable. For more information, see [Defining Returned Parameters](#).
- **If Multiple Records Match the Condition:** Options that apply to multiple matches:
 - Raise an Exception: Activates the exception exit. Enabled by default.
 - Save Number of Records to: Lists integer variables. The variable is set to the number of contact records to find.

Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds). Each prompt is interruptible.

In Visual IVR scripts, TTS prompts convert to text that appears in the body of the IVR window.

Page Tab

This tab contains the list of Visual IVR prompts shown in the customer's Web browser when a IVR script executes. Prompts on this tab display. Prompts listed in the Prompts tab do not display.

Note: If there are no prompts on this tab, the prompts in the Prompts tab display.

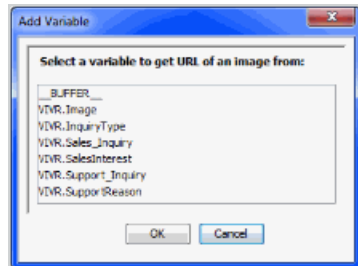
The screenshot shows the 'System Info Module Properties' dialog box with the 'Page' tab selected. The dialog has four tabs: General, Prompts, Page, and Text. The 'Page' tab contains two sections: 'Header' and 'Body'. Each section has a table with two columns: 'Type' and 'Text / File Name'. Below each table are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', and 'Convert to ML Prompt'. The 'Add' button for the 'Body' section is open, showing a dropdown menu with options: 'Add File', 'Add File from Variable', 'Add Text', and 'Add Multilingual Prompt'. At the bottom right of the dialog are 'Save' and 'Cancel' buttons.

You can add different types of prompts:

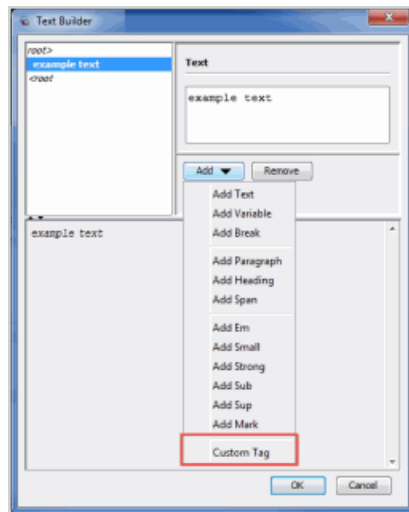
- **File:** URL of an image.



- **Variable:** Variable with the URL of an image.



- **Text:** Formatted text that can contain variables. You cannot add images. If you select a variable that contains the URL of an image, the URL displays.
- **Custom Tag:** See [Using the Visual IVR URL](#) for instructions on building custom tags.



In the Text Builder, you can use HTML to format text (bold, color, and emphasis) and you can insert variables. Unlike breaks in the TTS builder that are associated with speech, breaks in the Text Builder add a line or horizontal break. Click **Text** to preview the message.



Defining Search Conditions

This example searches for users with User Can Transfer Calls permission, who belong to the Inbound Sales queue group.

- 1 In the **General** tab, select the more menu next to the Lookup Condition field.
The Condition window displays.

Condition

Filter:

Condition Grouping: All

Expression:

Criteria #	Attribute	Operator	Value
------------	-----------	----------	-------

Add Remove Edit

Order by:

Rank	Contact Field	Order
------	---------------	-------

Add Remove Move Up Move Down

OK Cancel

2 Select **Add**.

3 Define the following filter criteria:

Condition

Attribute: Active

Operator: <

☒ Constant: true

☐ Variable: _BUFFER_

OK Cancel

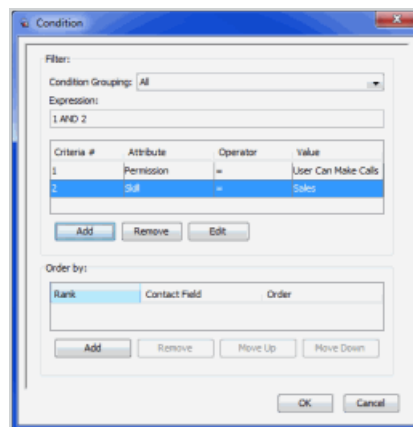
- **Attribute:** Element to compare with a constant or variable. For a list of attributes in each object, see the [table](#) below.
- **Operator:** Type of comparison. See [If/Else Module](#) for the comparison operations and compared value types.
- **Constant:** Available options vary depending on the attribute that you choose.

- **Variable:** List of call variables. For variables specific to this module, create a set of custom variables in the Variables tab of the IVR script properties window (see [Default Properties](#)).

4 Select **OK**.

The filter criterion item appears in the list.

5 Repeat these steps to add the Sales criteria.



6 In the **Condition Grouping** menu, select an option:

If you create more than one criteria, you can join them using logical operations.

- **Any:** Find objects that match any defined criteria.
- **All:** Find objects that match all defined criteria.
- **Custom:** Define a custom expression that describes how to use the criteria.

An expression automatically displays in the Expression field. For example, if you select All, the expression is 1 AND 2.

7 To define the sort order for the returned records, select **Add** (maximum three times) at the bottom.

- To select a different contact field, click the field.
- Select the order.

8 Define the sorting options:

The screenshot shows a 'Condition' dialog box. Under the 'Filter' section, 'Condition Groupings' is set to 'Any'. The 'Expressions' section contains two criteria: '1' with Attribute 'Permission' and Value 'User Can Make Calls', and '2' with Attribute 'Skill' and Value 'Sales'. Below this are 'Add', 'Remove', and 'Edit' buttons. The 'Order by' section has a table with three rows: Rank 1, Contact Field 'Extension', Order 'Ascending'; Rank 2, Contact Field 'Last Name', Order 'Ascending'; and Rank 3, Contact Field 'First Name', Order 'Ascending'. Below this table are 'Add', 'Remove', 'Move Up', and 'Move Down' buttons. At the bottom are 'OK' and 'Cancel' buttons.

9 Select OK.

These are the attributes and types of the variables for creating conditions for the system objects.

System Object	Attribute	Variable Type
Users	Active	Boolean (true or false)
	Extension (to route calls)	Number
	First Name	String
	Last Name	String
	Permission	String
	Role	String
	Skill	String
Skills	Name	String
Voice Mail	Campaign	String
	Creation date	Date
	Creation time	Time
	Extension	Number
	Is Deleted (boolean)	Number
	Is New (boolean)	Number
	Number (such as phone)	Number
Statistics: EWT	Module Name	Number

System Object	Attribute	Variable Type
(see Estimated Wait Time)		
Statistics: Skills	Range	String
	Skill	String
Statistics: Campaigns	Campaign	String
	Range	String

Defining Returned Parameters

For example, if a user account with a matching permission and skill is found, the user's extension is assigned to BUFFER.

- 1 On the General tab, select **Add**.

The **Edit Returned Parameter** window displays.

- 2 Select your desired parameters and variables:

- **Parameter:** Options that depend on the selected system object. For more information about parameters and data types, see the [table](#) below.
 - Users
 - Skills
 - Voicemails
 - Statistics: EWT (see [Estimated Wait Time](#))
 - Statistics: Skills
 - Statistics: Campaigns

- **Variable:** List of CAVs (call-attached variables). For variables specific to this module, create a set of custom variables in the Variables tab of the IVR script properties window (see [Default Properties](#)).

Use BUFFER if the previous module in the flow uses the value of the buffer, for example to route to the extension.

- 3 Select **OK**.

These are the parameters and variables for creating returned parameters for system objects.

System Object	Parameter	Variable Type
Users	Active	Boolean (<code>true</code> or <code>false</code> only)
	Extension (to route calls)	Number
	First Name	String
	Greeting (Path to greeting file)	String
	Last Name	String
	Number of new voicemail messages	String
	Number of saved voicemail messages	String
	PIN	
Skills	Name	String
Voicemails	Campaign	String
	Creation date	Date
	Creation time	Time
	Extension	Number
	File Name	String
	Is Deleted (boolean: 1 or 0)	Number
	or <code>true</code> or <code>false</code> ?	Number
	Is New (boolean: 1 or 0)	Number
	or <code>true</code> or <code>false</code> ?	Number
	Number (such as phone)	
Statistics: EWT (see Estimated Wait Time)	EWT	Number
Statistics: Skills	Agents Logged In	Number
	Agents On Call	Number
	Agents Ready for Calls	Number
	Cumulative Longest Queue Time	Number
	Calls In Queue	Number
	Longest Queue Time	
	Queue Callbacks	
Statistics: Campaigns	Avg Handle Time	Number
	Avg Speed of Answer	Number
	Calls Abandoned	Number

System Object	Parameter	Variable Type
	Drop Call %	Number
	First Call Resolution	Number
	Service Level (Queue)	Number
	Total Calls	Number

Third Party Transfer Module

This module enables you to transfer a call to a phone number outside your VCC.

The screenshot shows the '3rd Party Transfer Module Properties' dialog box with the 'General' tab selected. The dialog includes a description: 'The 3rd-Party Transfer Module can be used to blind-transfer (single-step transfer) a caller to a specified phone number.' Below this, there are fields for 'Module Name' (containing 'ThirdPartyTransfer3') and 'Third Party Number'. The 'Third Party Number' section has two radio buttons: 'Constant (Example: 5101232341)' which is selected, and 'Variable'. Under the 'Recording' section, there are checkboxes for 'Record 3rd Party call' and 'Return After 3rd Party Call'. Below these are several input fields: 'Send Data to 3rd Party', 'Receive Data from 3rd Party', 'Max Time of 3rd Party Call (Seconds)' (set to 300), 'Digit to Interrupt 3rd Party Call & Return', and 'Ringing Timeout (Seconds)' (set to 10). At the bottom right are 'Save' and 'Cancel' buttons.

Note

Error response handling while processing third party transfer actions are supported.

General Tab

- **Third Party Number:** Phone number for the transfer:
 - Constant: Entry field for the transfer phone number. Do not enter a one (1) in front of the number. Enter the number without dashes, parentheses, or other characters. You must enter phone numbers according to your domain and VCC configuration:

- If your domain is enabled for the E.164 format, you may need to enter phone numbers in your national format or phone numbers up to 15 digits, starting with the plus (+) sign and the country code. See your administrator.
 - If your domain is not enabled for the E.164 format, you must enter 10 digits to conform to the [North American Numbering Plan](#) (NANP).
 - Variable: list of contact, agent, system, call, IVR, and user (CAV) variable that contains the voice mailbox number. This menu contains string variables only.
- **Record 3rd Party call:** Whether to record calls transferred to a third party.
If **Set Recording Name to Session ID** is selected in the General tab of the campaign settings, third-party transfer recordings are uploaded in this file format:
`<callSessionId>@<timeOfCreate><IVRModuleName>.ext`
You are responsible for notifying the third party that the call is being recorded and for obtaining consent when required by law.
Select this option to enable third party data transfer.
- **Return After 3rd Party Call:** Whether to return the call to your domain after the third-party interaction is completed.
Select this option to enable third party data transfer.
Data transfer requires additional configuration. Contact your Five9 representative.
- **Send Data to 3rd Party:** Enter KVList of variables to send to third party.
- **Receive data From 3rd Party:** Enter return mapping for KVList of variables from third party.
- **Max Time of 3rd Party Call (seconds):** Configure up to 3600 seconds (sixty minutes) duration for third party calls using this transfer module.
- **Digit to Interrupt 3rd Party Call and Return:** Configure dialing pad key (0-9,*,#) to interrupt the third-party call and return that interaction to your contact center.
- **Ringing Timeout (seconds):** Configure up to 30 seconds for maximum allowable ring time.

Dispositions Tab

The default disposition is **Transferred to third party**. For more information, see [Selecting Dispositions](#).

Update System Info Module

The Update System Info module updates the currently selected system object.

Because you may not use call variables that contain sensitive information in this field, these variables do not appear in the menu even if they exist in the Call Variables folder of the VCC.

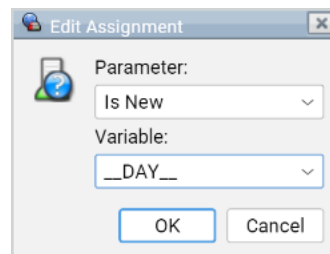
The screenshot shows the 'Update System Info Module Properties' dialog box with the 'General' tab selected. The dialog contains the following fields and controls:

- Module Name:** A text box containing 'SystemUpdate3'.
- System Object:** A dropdown menu with 'Voicemails' selected.
- Found in Module:** A dropdown menu.
- Assignments:** A table with two columns: 'Parameter' and 'Variable/Value'. Below the table are 'Add', 'Remove', and 'Edit' buttons.
- Save and Cancel:** Buttons at the bottom right of the dialog.

Text inside the dialog: The Update System Info module can be used to modify parameters of a system object that is selected by a specified System info module.

- **System Object:** Type of system objects: Users and Voicemails.
- **Found in Module:** List of System Info modules.

- **Assignments:** System parameters that you can update with a variable:
 - Users: You can modify the PIN.
 - Voicemails: Parameters that you can update with a variable:
 - Is New: contact, system, agent, IVR, call, and user call variables.
 - Is Deleted: contact, system, or user call variable.



Voice Input Module

After callers are prompted for information, the Voice Input module saves the speech as a recording. The recording stops when one of these conditions is met:

- The duration of the final silence is exceeded.
- A DTMF key is pressed.
- The maximum recording time is exceeded.
- The caller hangs up.

If configured accordingly, the module can play the recording to callers for confirmation, with the option to re-record the message if necessary. The Skill Transfer and Agent/Voice Mail Transfer modules have a property that controls how the recorded files are presented and stored. You can set the Action for Recorded Files option (General tab):

- Keep as Recording: an inbound campaign can collect callers' address information. The recorded file becomes available in reporting.
- Send to Agent: callers can be transferred to an agent with the Skill Transfer Module. The recorded file becomes available in the agent desktop and in reporting.

You can assign some recording characteristics to IVR script variables. You can save the following information about a recording. For more information, see [Using Recorded Files](#).

When using Visual IVR scripts, you may also add images to pages and menus to enhance your brand or to describe a product or service. However, you cannot add links to images. For more information, see [Page Tab](#).

[General Tab](#)

[Prompts Tab](#)

[Events Tab](#)

[Confirmation Tab](#)

[Page Tab](#)

[Dispositions Tab](#)

The screenshot shows the 'Voice Input Module Properties' dialog box with the 'General' tab selected. The dialog has tabs for General, Prompts, Events, Confirmation, Page, Text, and Disposition. A description states: 'The Voice Input Module can be used to capture a caller's speech and save it as a call recording or voicemail, or deliver it to an agent.' The 'Module Name' field contains 'Tell us your interest'. 'Max Time (secs)' is set to 30 and 'Final Silence (msecs)' is set to 3,000. Checkboxes for 'Use DTMF to Terminate Recording' and 'Save User Input as Recording' are checked. Under 'Post-Recording Analysis', several options are unchecked. The 'Use the Recording as a Greeting for the Extension' option is selected with the 'Variable' radio button. 'Max Attempts' is set to 1, and the 'Collapsible' checkbox is unchecked. 'Save' and 'Cancel' buttons are at the bottom right.

Property	Value
Module Name	Tell us your interest
Max Time (secs)	30
Final Silence (msecs)	3,000
Use DTMF to Terminate Recording	Checked
Save User Input as Recording	Checked
Duration of the Recording (msecs)	Unchecked
Variable to Access the Recording	Unchecked
Save Pressed Termination Key to	Unchecked
Generate Exception if Max Time Was Reached	Unchecked
Use the Recording as a Greeting for the Extension	Checked (Variable)
Max Attempts	1
Collapsible	Unchecked

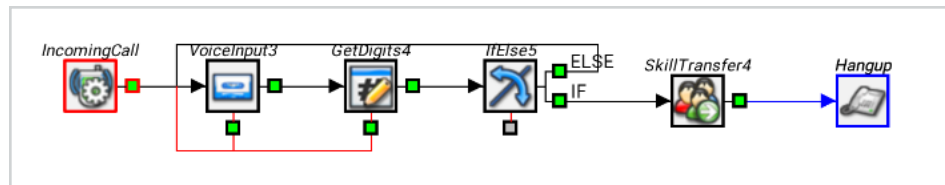
General Tab

- **Module Name:** Name of the module that represents the role of the module in the IVR script. The name also applies to the call recordings in the agent application and in reports.

In Visual IVR scripts, the name is converted to text that appears in the heading of the IVR window.

- **Max Time:** Number of seconds allowed for the recording. The default is 30 seconds, excluding the prompt time. After that time, you may receive a NoMatch exception.
- **Final Silence:** Number of milliseconds that indicates the end of speech. The default is 3000 ms (3 seconds).
- **Use DTMF to Terminate Recording:** Whether to allow the caller to press any key to stop the recording.
 - Enabled: by default, stop the recording when the caller presses a key. Required if you want to use the **Save Pressed Termination Key to** option.
 - Disabled: do not stop the recording.
- **Save User Input as Recording:** Whether to save the caller's input as recorded files.;
 - Enabled: by default, save the caller's input as a recording.
 - Disabled: do not save the caller's input.
- **Duration of the Recording:** Number of milliseconds that indicates the length of the recording. Use a call-attached variable that is an integer.
- **Variable to Access the Recording:** Variable that contains the recording. Be sure to select a variable if you want to use confirmation with DTMF input in the Confirmation Tab shown below. Speech recognition must be configured for your account.

Use other modules, such as Get Digits or Play, to confirm the input. For example, the Voice Input Module prompts callers to say their name, which is recorded. The next module, Get Digits, prompts callers to confirm the information or to record it again. After the information is confirmed, callers are transferred to an agent.



- **Save Pressed Termination Key to:** **Use DTMF to Terminate Recording** must be enabled.

String variable that contains the DTMF key, such as # (pound) that terminates the recording if pressed by the caller. Otherwise, the variable is assigned to empty.

- **Generate Exception if Max Time Was Reached:** Whether to generate an exception when the number of seconds allowed for the recording (Max Time value) is reached when the IVR goes to the module connected to the Exception branch or to the next module if no exception branch exists.
 - Enabled: The system generates an exception.
 - Disabled: The system does not generate an exception but generates a NoMatch event.
- **Use the Recording as a Greeting for the Extension:** Replaces an existing greeting for a specified extension with the created recording:
 - Constant: up to six-digit extension number.
 - Variable: List of contact, agent, system, call, IVR, and user (CAV) variable that contains the voice mailbox number. This menu contains string variables only.

You may use this option to enable agents to play a personalized voice mailbox greeting when callers are transferred to agents.

- **Max Attempts:** Number of possible attempts to capture the caller's speech before leaving the module with error code. The default value is 1. To allow callers to re-record, the value must be greater than 1.
- **Collapsible:** Visual IVR feature that applies to all types of prompts. Whether to concatenate all the prompts in the module whether they are played (standard prompts) or displayed (Visual IVR prompts) to callers.

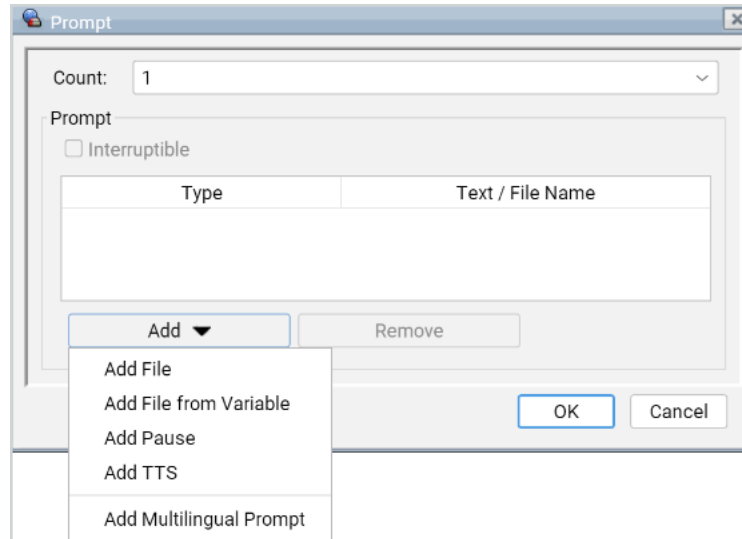
Voice Input

- Enabled: Concatenate prompts. Callers see at once all the prompts in the same window.
- Disabled: Do not concatenate prompts. Callers hear or see each prompt after performing the requested action.

Prompts Tab

List of default and custom prompts in the Prompts folder. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in any sequence. Between prompts, you can add brief pauses (milliseconds).

In Visual IVR scripts, TTS prompts are converted to text that appears in the body of the IVR window.



- **Count:** Minimum number of times that the user must speak for the prompt to be played. The default is 1.
- **Interruptible:** Not in use.

Events Tab

List of event handling options. You can add one or more prompts from a file, a variable, or from the TTS builder. You can arrange the prompts in a specific sequence. The list contains the default messages:

Event	Default Messages	Action
NoMatch	Your recording is too long. Please repeat.	Continue
NoInput	I cannot hear you.	Reprompt

- **Event:** List of possible events:
 - NoMatch: Generated if the recording stopped because it exceeded the maximum allowed recording duration and, in the General tab, **Generate Exception If Max Time Was Reached** is disabled.
 - NoInput: Generated if the recording time is zero.
 - Help: Generated if the caller presses the Help button on the keypad.
- **Count:** Number of events (1–11) for which the action should be taken.

- **Action:** Action taken when the specified event occurs.
 - Reprompt: Repeat the prompt listed in the Prompts tab.
 - Continue: Proceed to next attempt without playing the prompt listed in Prompts tab.
 - Exit: Terminate the script.
- **Interruptible:** Not in use.

Confirmation Tab

This tab appears only if speech recognition is configured for your account. For confirmation with DTMF input, you must have selected **Variable to Access the Recording** in the General tab.

This tab, which accepts DTMF input, such as *Press 1*, or speech input, such as *Yes*, enables you to set how messages are recorded again if **Max Attempts** in the General tab is greater than 1. After a recording is played, the caller accepts or rejects the recording:

- **Accepts:** The IVR proceeds to the next module.
- **Rejects:** The caller is prompted to record the message again. The prompt to record again is specified as NoMatch events.

The screenshot shows the 'Voice Input Module Properties' dialog box with the 'Confirmation' tab selected. The 'Confirmation' section has 'Not Required' selected. The 'Confirmation Dialog' section has 'Max Attempts to Confirm' set to 3 and 'No Input Timeout (secs)' set to 3. The 'Prompt' section contains the text 'Your message is: To keep it, say "Yes" or press "1". To re-record the message, say "No" or press "2".' and an 'Edit' button. The 'Interruptible' checkbox is unchecked. The 'Events' section contains a table with three columns: Event, Event's Prompt, and Action.

Event	Event's Prompt	Action
NoMatch	NoMatchPrompt	reprompt
NoInput	NoInputPrompt	reprompt

At the bottom of the dialog are 'Save' and 'Cancel' buttons.

- **Confirmation:** Whether to use the options on this tab:
 - **Required:** Callers can listen to the message to confirm that it is correct or to record it again. Select this option to activate the confirmation options.
 - **Not Required:** By default, callers cannot listen to the recorded message.
- **Max Attempts to Confirm:** Number of attempts allowed to accept the recorded message. The default is 3.
- **No Input Timeout:** Number of seconds allowed before the NoInput event occurs. The default is 3.
- **Prompt:** Confirmation prompt played to callers. You can use the default prompt or create a prompt specific to your needs.
- **Events:** Actions that can occur during confirmation and re-recording, for example with no match, no input, or help request events. You can customize the event in this tab and add new events in the Events tab. The list contains these default messages.

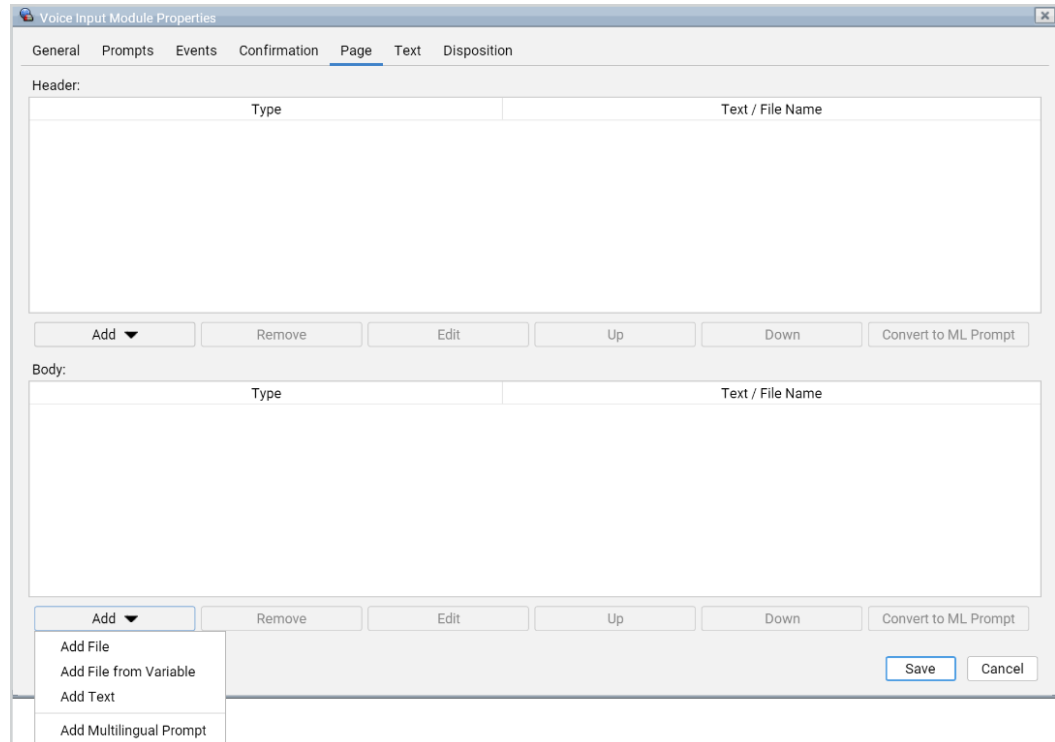
Event	Message	Action
NoMatch	Incorrect input.	Continue
NoInput	I cannot hear you.	Reprompt

Page Tab

This tab contains the list of Visual IVR prompts that are shown visually in the customer's Web browser when a IVR script is executed.

If this tab contains prompts, these are shown whereas prompts in the Prompts tab are not.

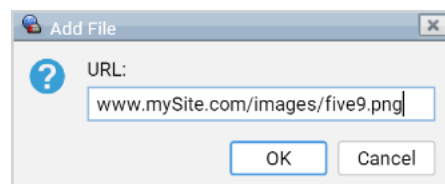
If this tab does not contain prompts, the prompts in the Prompts tab are shown.



The image shows the 'Voice Input Module Properties' dialog box with the 'Page' tab selected. The dialog has tabs for General, Prompts, Events, Confirmation, Page, Text, and Disposition. The 'Page' tab contains two sections: 'Header' and 'Body'. Each section has a table with two columns: 'Type' and 'Text / File Name'. Below each table are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', and 'Convert to ML Prompt'. The 'Add' button for the 'Body' section is open, showing a dropdown menu with options: 'Add File', 'Add File from Variable', 'Add Text', and 'Add Multilingual Prompt'. At the bottom right of the dialog are 'Save' and 'Cancel' buttons.

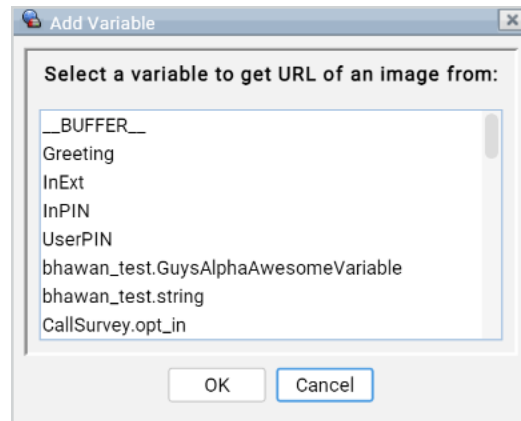
This tab enables you to add different types of prompts:

- **File:** URL of an image.



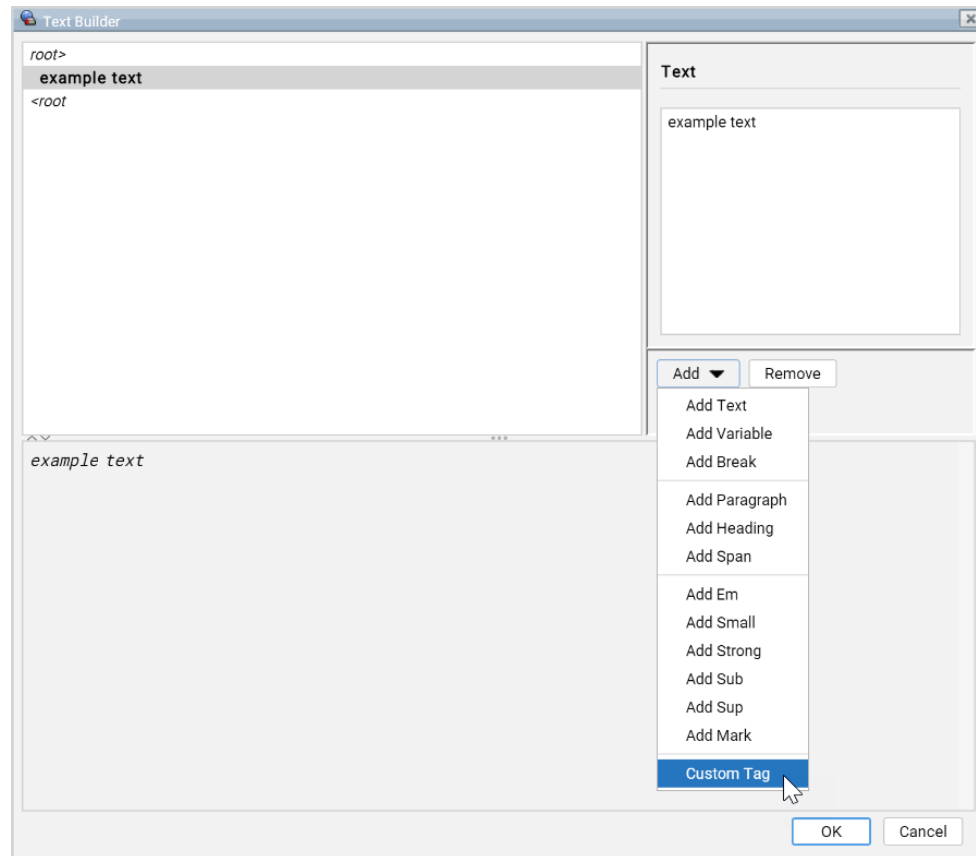
The image shows the 'Add File' dialog box. It has a title bar with a question mark icon and the text 'Add File'. Inside, there is a label 'URL:' followed by a text input field containing the URL 'www.mySite.com/images/five9.png'. At the bottom are 'OK' and 'Cancel' buttons.

- **Variable:** Variable that contains the URL of an image.



- **Text:** Formatted text that can contain variables. You cannot add images. If you select a variable that contains the URL of an image, the URL is displayed.

- **Custom Tag:** For detailed usage information, see [Using the Visual IVR URL](#).



The text editor enables you to create HTML text that contains formatted text and variables. You can customize the text with bold, color, and emphasis. Unlike breaks in the TTS builder, which are associated with speech, breaks in this text editor add a line or horizontal break.

The Test button enables you to preview the HTML text.



Dispositions Tab

The default disposition is Caller Disconnected. For more information, see [Selecting Dispositions](#).

Voicemail Transfer Module

The VoiceMail Transfer module enables you to transfer recordings captured with an IVR to a personal or skill group voice mailbox. By default, a message remains in the module for 59 minutes and 59 seconds. Because transferring a call ends the IVR script, you need to specify dispositions.

If the caller does not leave a voice message, use another type of exit to route the call.

The screenshot shows a window titled "Voicemail Transfer Module Properties" with two tabs: "General" and "Disposition". The "General" tab is active. It contains a description: "The **Voicemail Transfer Module** can be used to transfer recordings captured in the IVR to a Personal or Skill voicemail box." Below this, there is a "Module Name" field with the value "VoiceMailTransfer3". The "Voicemail Box Type" section has two radio buttons: "Personal" (selected) and "Skill". The "Voicemail Box" section has two options: "Constant" (selected) and "Variable", each with a dropdown menu. At the bottom, there are two checkboxes: "Add Voice Input Files to Voicemail" (unchecked) and "Create Voicemail Only From Voice Input Files" (unchecked). "Save" and "Cancel" buttons are at the bottom right.

General Tab

- **Voicemail Box Type:** Select the type of voice mailbox for the message from the caller: Personal or Skill.
- **Voicemail Box:** Name or variable that specifies the name of the voice mailbox.
 - **Constant:** depending on the Voicemail Box Type that you selected, this menu contains the list of possible users or skills.
Variable: list of contact, agent, call, IVR, and user (CAV) variable that you can use to find the voice mailbox. This menu contains string variables only.
Because you may not use call variables that contain sensitive information in this field, these variables do not appear in the menu even if they exist in the Call Variables folder of the VCC.
- **Add Voice Input Files to Voicemail:** Whether to transfer the files of the recordings with the voicemail messages.
 - **Enabled:** transfer recordings. To transfer recordings to a personal or skill voice mailbox, include the Voice Input module in the script.

- Disabled: recorded files are not transferred to agents or skills; saved recordings can be accessed only from reports. See also [Using Recorded Files](#).
- **Create Voicemail Only From Voice Input Files:** Whether to transfer only recorded files as voicemail to the specified voice mailbox.

You can use the [Voice Input Module](#) to limit the length of voicemail messages received. Using the Voice Input module will also bypass an agent's personal greeting.

Dispositions Tab

The default disposition is Sent to Voicemail. For more information, see [Selecting Dispositions](#).

IVR Examples

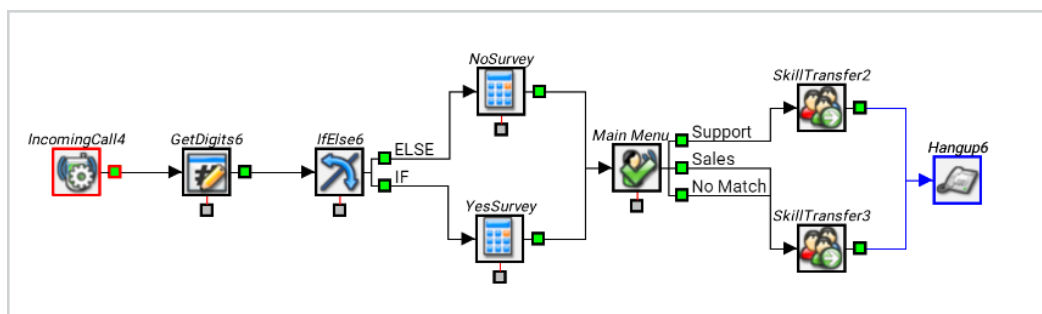
These topics present example scripts using several modules types.

[Simple Campaign Script for Post-Call Surveys](#)
[Complex Campaign Script for Post-Call Surveys](#)
[Creating an IVR Script](#)
[Creating an IVR Loop](#)
[Configuring the Query Module](#)
[Creating an IVR Script with XML Web Service Query](#)
[Creating an IVR Script with Queue Callback](#)
[Creating an IVR Script for Tracking a Shipment](#)
[Using an Agent Extension in an IVR](#)
[Incoming Call Transfer to Studio IVA](#)
[Managing Voicemail](#)
[Creating an Opt-Out IVR Script](#)
[Flowcharts](#)

Simple Campaign Script for Post-Call Surveys

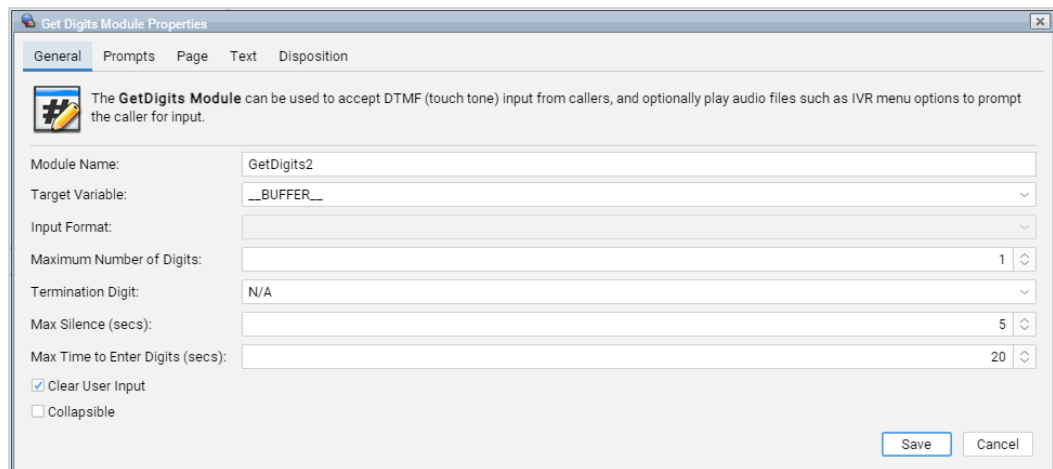
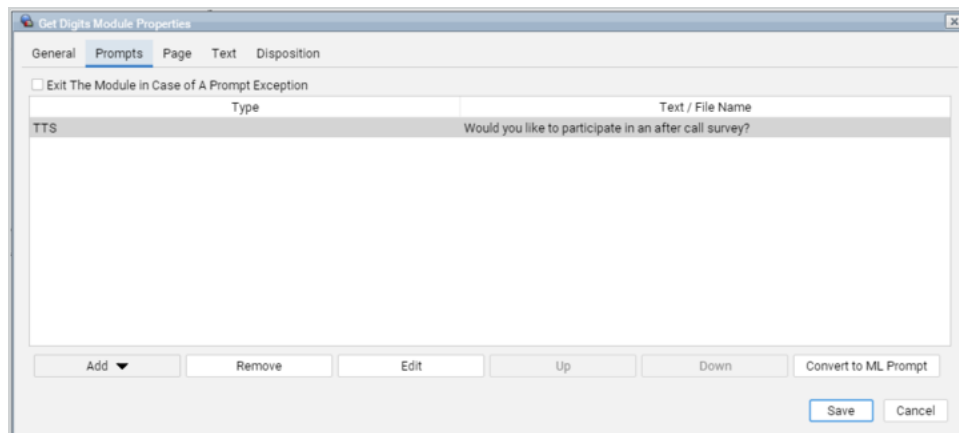
This example describes how to create a script to transfer calls to a survey campaign.
See also [Complex Campaign Script for Post-Call Surveys](#).

[Get Digits Module](#)
[If/Else Module](#)
[Set Variable Module \(IF Branch\)](#)
[Set Variable Module \(ELSE Branch\)](#)



Get Digits Module

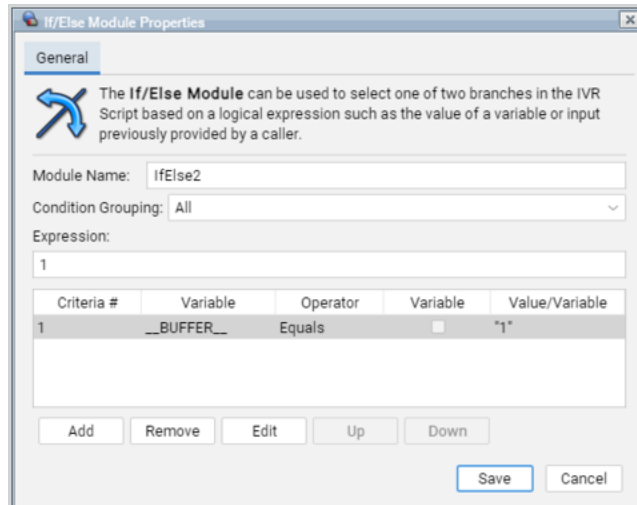
This module asks the caller to participate in an after-call survey. The maximum number of digits in the BUFFER variable is 1. The caller is asked to enter 1 to start the survey or 2 to skip the survey.



If/Else Module

Check the value of the BUFFER variable:

- BUFFER=1: (start the survey) proceed to **Set Variable Module (IF Branch)**.
- BUFFER=2: (skip the survey) proceed to **Set Variable Module (ELSE Branch)**.



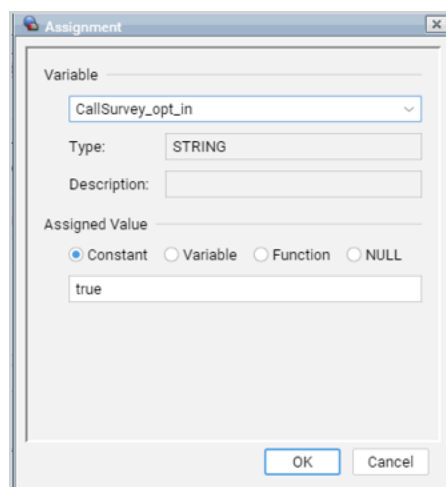
Set Variable Module (IF Branch)

Configure the module as follows:

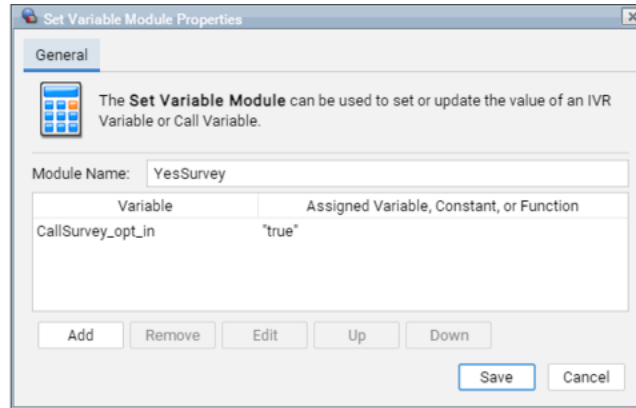
- 1 Assign a value to the `CallSurvey.opt_in` variable.

Although it is a boolean variable, the data type that you see is `string`.

- 2 Select **Constant** and enter `true`.



- 3 Click **OK**.

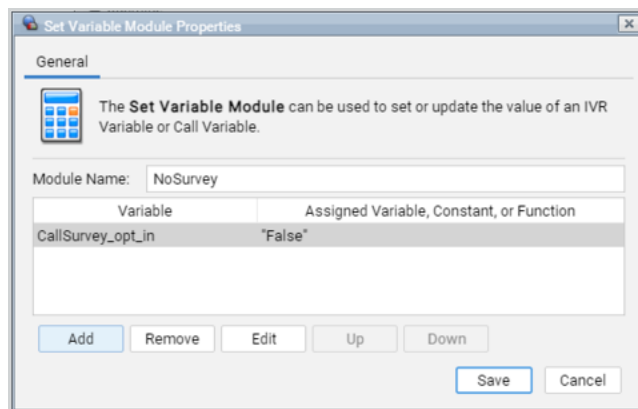


- 4 Proceed with the survey.

Set Variable Module (ELSE Branch)

Configure the module as follows:

- 1 Assign a value to the `CallSurvey.opt_in` variable.
- 2 Select **Constant** and enter `false`.



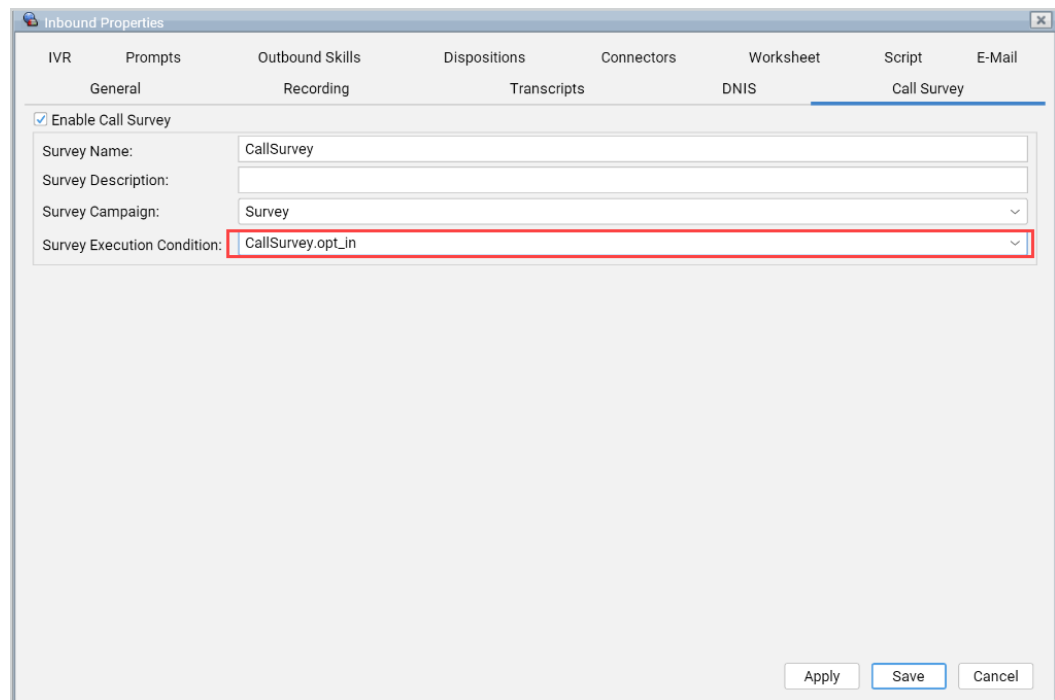
- 3 Click **Save**.

Complex Campaign Script for Post-Call Surveys

This section describes how to configure an IVR script for call surveys in which calls are transferred at the end of a campaign.

[Call Variables](#)
[IVR Script](#)

The `CallSurvey.opt_in` variable was added to an inbound campaign as call survey execution condition:



The screenshot shows the 'Inbound Properties' dialog box with the 'Call Survey' tab selected. The 'Enable Call Survey' checkbox is checked. The 'Survey Name' is 'CallSurvey', 'Survey Description' is empty, and 'Survey Campaign' is 'Survey'. The 'Survey Execution Condition' is 'CallSurvey.opt_in', which is highlighted with a red border. The 'Apply', 'Save', and 'Cancel' buttons are at the bottom right.

IVR	Prompts	Outbound Skills	Dispositions	Connectors	Worksheet	Script	E-Mail
General	Recording	Transcripts	DNIS	Call Survey			

☒ Enable Call Survey

Survey Name: CallSurvey

Survey Description:

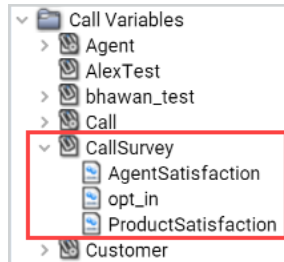
Survey Campaign: Survey

Survey Execution Condition: CallSurvey.opt_in

Apply Save Cancel

Call Variables

There is a Call Survey Variable group. It is not available by default. You have to create it by right-clicking **Call Variables** and selecting **Add Call Variable Group**. In this example it is called `CallSurvey`. The variables created here are used for questions in the following Call Survey script example.

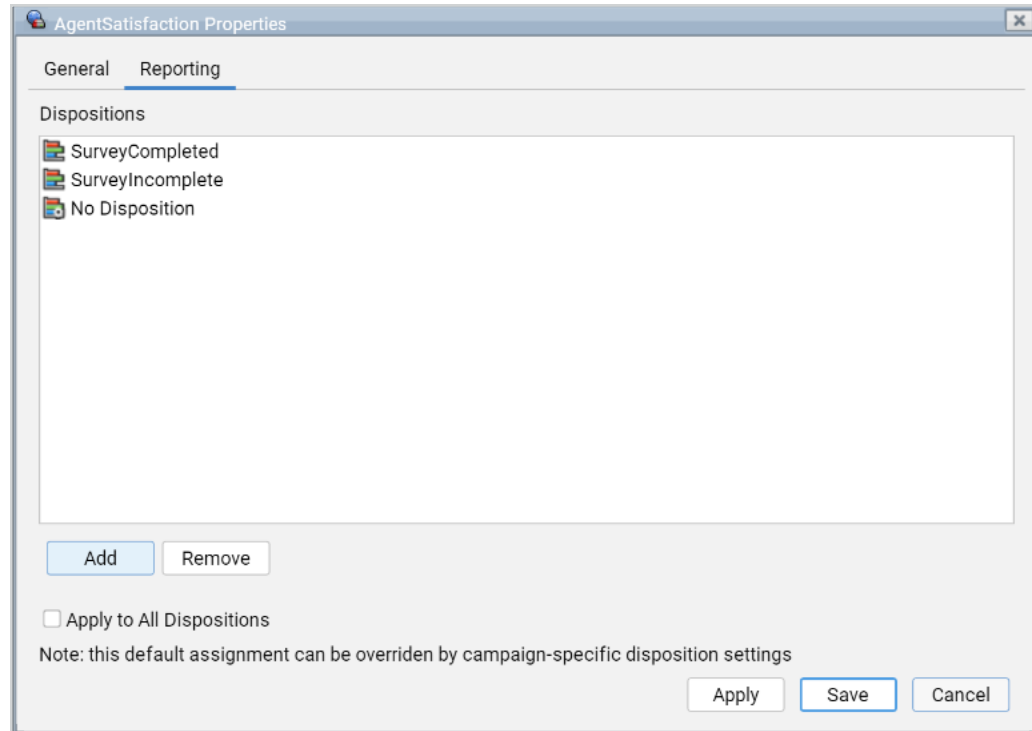


Read more in [Creating Call Variable Groups](#), [Creating Call Variables](#).

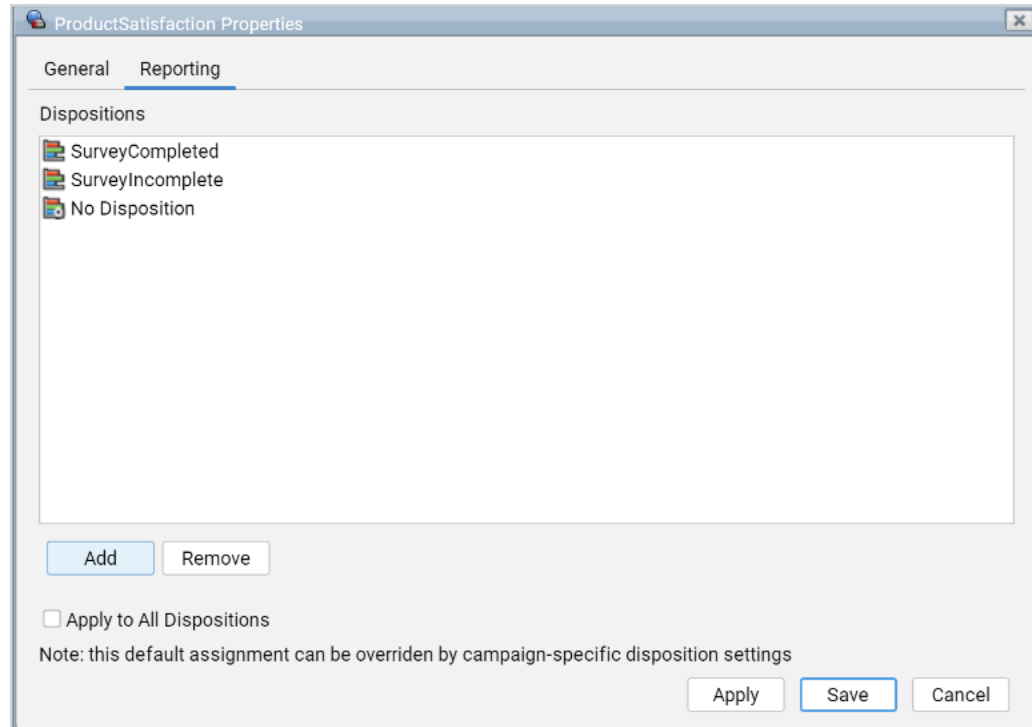
Agentsatisfaction Variable. You can enable **Reporting Call Variable** for your domain if you plan to use it. You can choose whether or not report on this variable (it is saved or removed).

The 'AgentSatisfaction Properties' dialog box is shown with the 'Reporting' tab selected. The variable name 'AgentSatisfaction' is in the top text box. Below it is a 'Description:' label and an empty text box. The 'Reporting Call Variable (Saved in Database)' checkbox is checked. The 'Field Contains Sensitive Data' checkbox is unchecked. The 'Data Type' is set to 'Number'. The 'Digits before decimal point' is set to 1, and the 'Digits after decimal point' is set to 0. Under the 'Restrictions' tab, the 'Required' and 'Predefined List' checkboxes are unchecked. The 'Min Value' checkbox is checked with a value of 1, and the 'Max Value' checkbox is checked with a value of 9. At the bottom right are 'Apply', 'Save', and 'Cancel' buttons.

You can choose only the applicable dispositions for you call survey script. In this example we use only SurveyCompleted and SurveyIncomplete dispositions, which were created earlier.

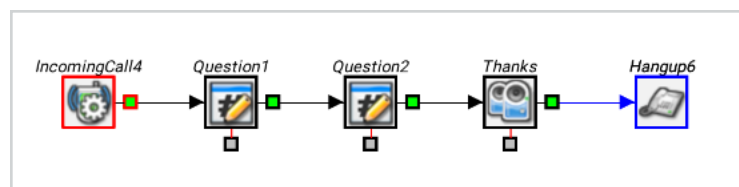


Productsatisfaction Variable. We have SurveyCompleted and SurveyIncomplete dispositions set in here.

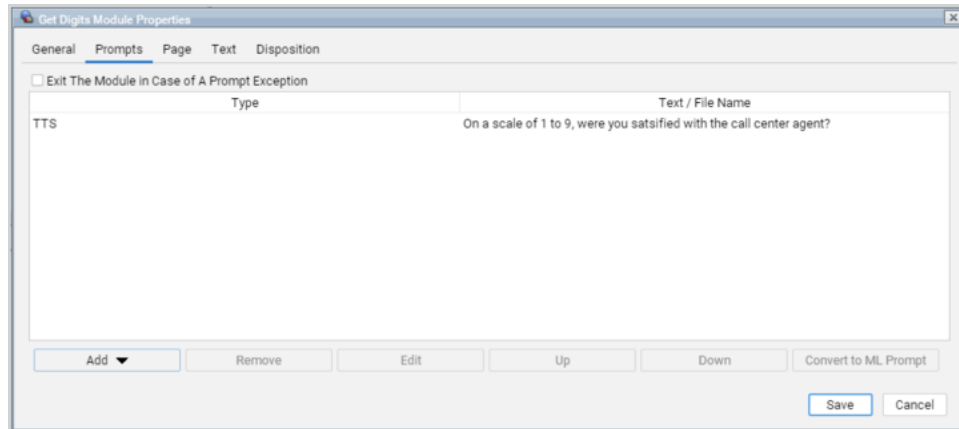


IVR Script

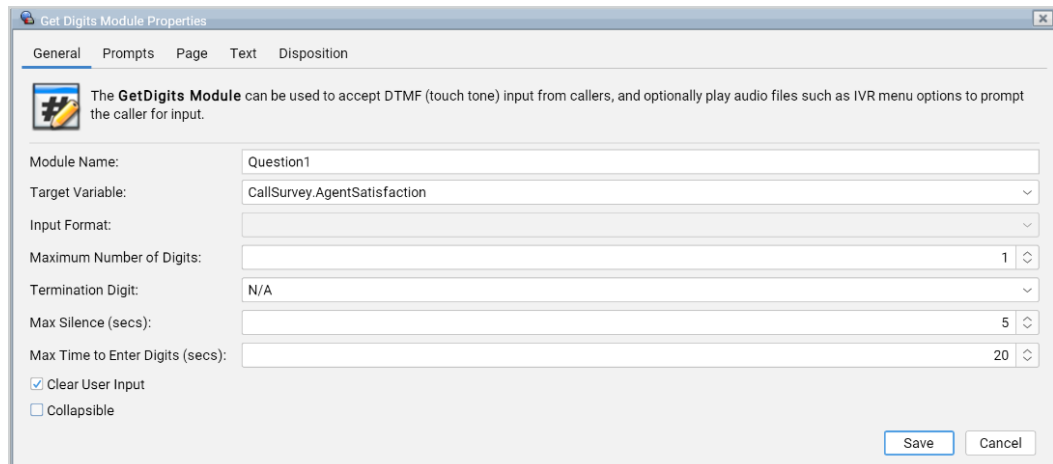
There is the Incoming Call Module and there may be a series of questions after it. There are 2 questions in this example.



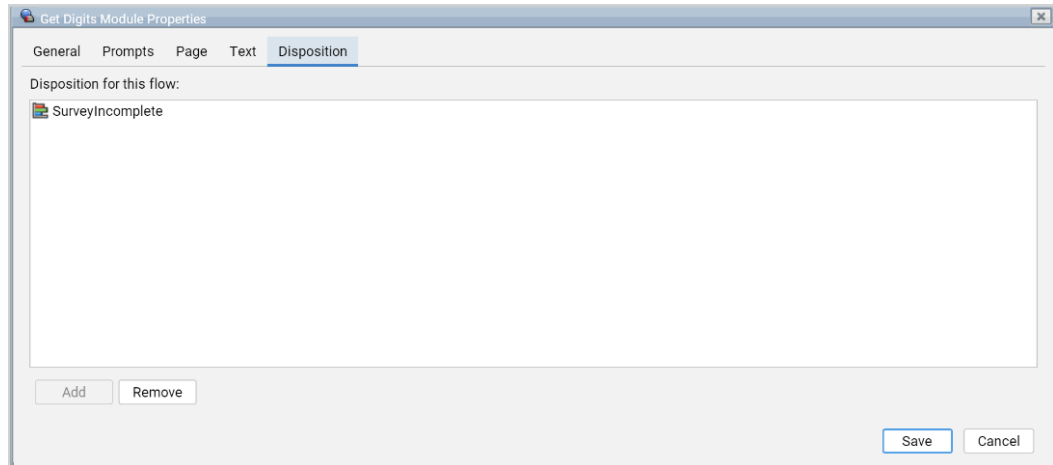
Get Digits Module 1. The first question asks the caller: *On a scale of 1 to 9, were you satisfied with the call center agent?*



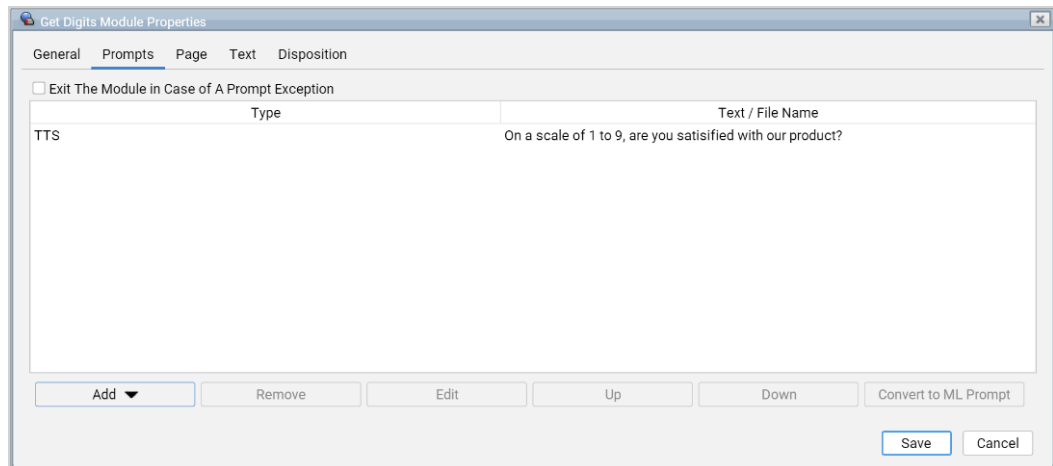
On the General tab, the target variable is a created variable AgentSatisfaction. The maximum number of digits is 1 because we have asked on the scale of 1 to 9. There is no termination digit unless you want the caller to confirm it.



The disposition configured here is SurveyIncomplete. It tells us whether or not the caller actually went through the entire Call Survey script.



Get Digits Module 2. The next question asks: *On a scale of 1 to 9, are you satisfied with our product?*



The disposition is SurveyIncomplete, the target Variable is ProductSatisfaction, and the maximum number of digits is 1.

The screenshot shows the 'Get Digits Module Properties' dialog box with the 'General' tab selected. The dialog contains the following fields and options:

- Module Name:** Question2
- Target Variable:** CallSurvey.ProductSatisfaction
- Input Format:** (empty)
- Maximum Number of Digits:** 1
- Termination Digit:** N/A
- Max Silence (secs):** 5
- Max Time to Enter Digits (secs):** 20
- ☒ Clear User Input
- ☐ Collapsible

Buttons at the bottom right: Save, Cancel.

Play Module. We thank the caller for calling by using the Play module.

The screenshot shows the 'Play Module Properties' dialog box with the 'Disposition' tab selected. The dialog contains the following elements:

- Disposition for this flow:** SurveyCompleted
- Buttons: Add, Remove
- Buttons at the bottom right: Save, Cancel

Hangup Module. The disposition is SurveyCompleted.

Creating an IVR Script

In this example, a dialer calls to the inbound number assigned to the campaign, is greeted with a message, and given the choice to transfer to Sales or Support by pressing 1 or 2. If the maximum hold time exceeds one minute, the caller is transferred to voicemail. If the caller makes an invalid entry (a number other than 1 or 2), the

choices are repeated. If the caller makes five consecutive invalid entries, the call is disconnected. This example uses two skill groups, Sales and Support, and a custom greeting, such as: *Thank you for calling <your company name>. Press 1 for Sales or 2 for Support.*

[Adding and Configuring the Play Module](#)

[Adding and Configuring the Get Digits Module](#)

[Routing Users to Skill Groups](#)

[Adding the Hang Up Module](#)

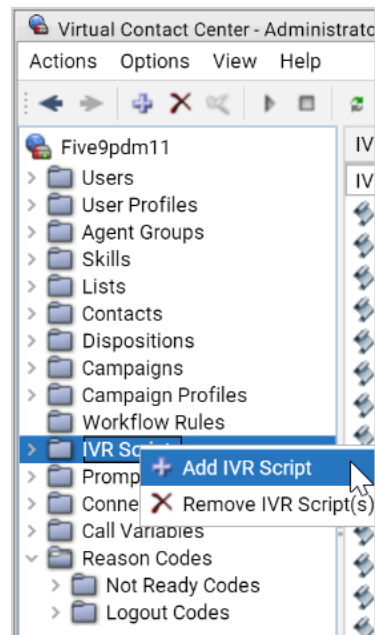
[Adding and Configuring the If/Else Module](#)

[Adding and Configuring the Set Variable Module](#)

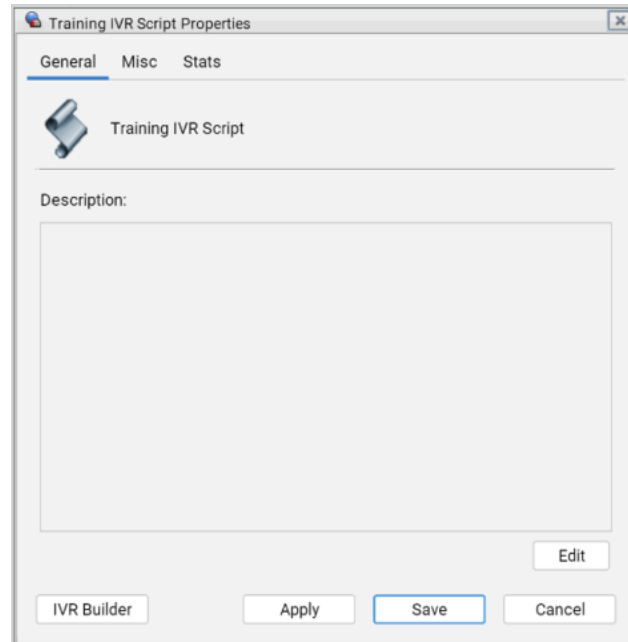
[Linking From the If/Else Set Variables Modules](#)

To create an IVR script:

- 1 In the Navigation pane, right-click IVR Scripts, and select **Add IVR Script**.



- 2 Enter the name of the IVR Script.
- 3 Select **OK**.
- 4 To start creating the script flow, open the **Script Properties**.
- 5 Double-click the script name or right-click and select **View Properties**.
- 6 Describe the script and its campaigns.

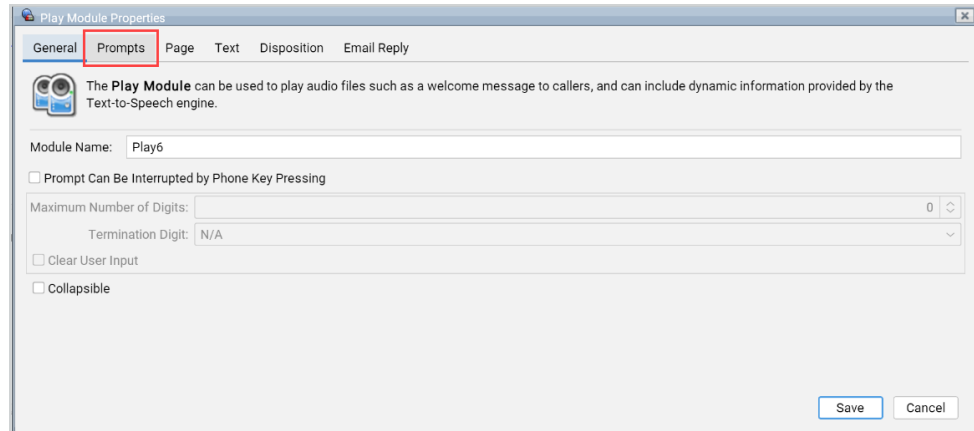


- 7 To open the Visual IVR Script designer, click **Edit**.
By default, the working area contains the Incoming Call module.

Adding and Configuring the Play Module

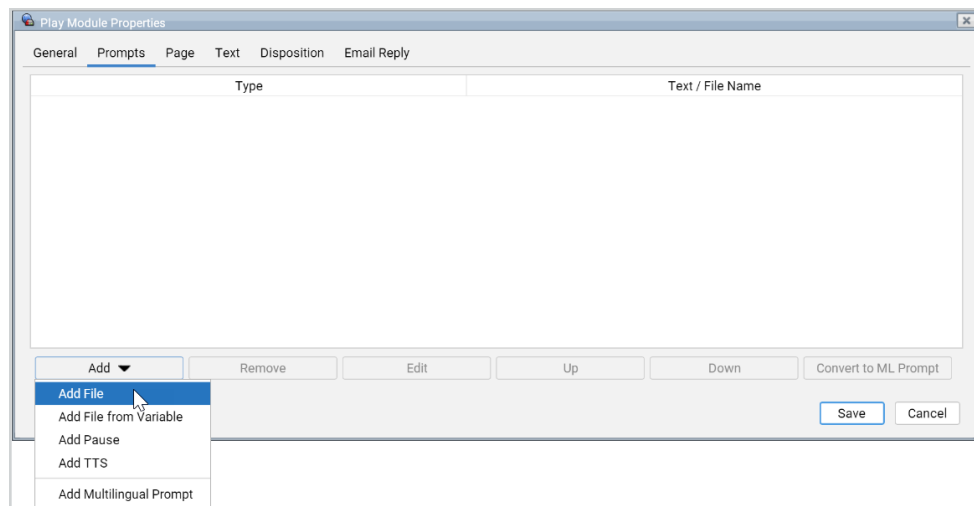
In the example, customers receive a welcome greeting and further instructions when they call the inbound campaign and next choose the skill group by entering 1 or 2.

- 1 Double-click the Play module.
- 2 Select the **Prompts** tab.



You can create voice messages by using the preloaded prompts or the Text-to-Speech functionality. If you plan to use several prompts, add a pause between each.

- 3 To add the prompt, click **Add > Add file**.

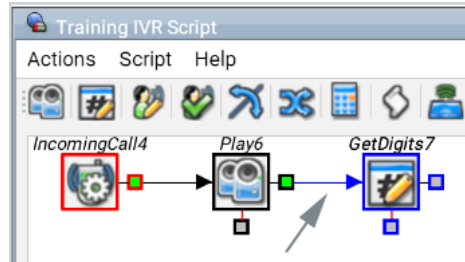


To create a voice message with TTS, select **Add. > Add TTS**.

- 4 Select a prompt from the list.

Adding and Configuring the Get Digits Module

- 1 Drag the Get Digits module icon from the toolbar to the working area.
- 2 Connect the Play and Get Digits module by using the right connector.



- 3 Double-click the Get Digits module.
- 4 In the **General** tab, set these options.
 - a Set the **Maximum Number of Digits** to 1 to ensure that customers enter only one number.
 - b Select **Clear User Input** so that all input from the customers clears when the call reaches this module.

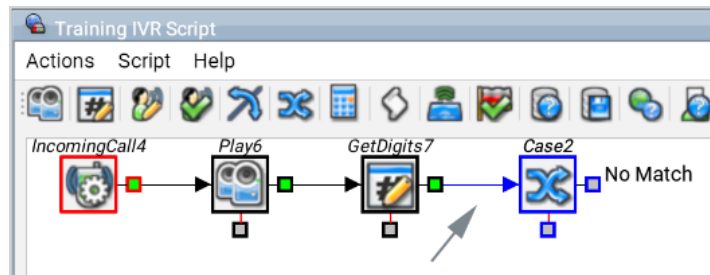
The screenshot shows the "Get Digits Module Properties" dialog box with the "General" tab selected. The description states: "The GetDigits Module can be used to accept DTMF (touch tone) input from callers, and optionally play audio files such as IVR menu options to prompt the caller for input." The fields are: Module Name: GetDigits7; Target Variable: __BUFFER__; Input Format: (empty); Maximum Number of Digits: 1 (highlighted with a red box); Termination Digit: #; Max Silence (secs): 2; Max Time to Enter Digits (secs): 20. The "Clear User Input" checkbox is checked (highlighted with a red box), and "Collapsible" is unchecked. "Save" and "Cancel" buttons are at the bottom right.

- 5 Select **Save**.

Routing Users to Skill Groups

You must route user input to the skill groups. Use the Case module to split the script flow based on user input to the appropriate branches. By default, this module has a No Match branch for when the user input does not match the criteria defined in the Case module (see [Configuring the No Match Branch](#)).

- 1 Add the Case module and connect it from the Get Digits module.



Route the user to skill groups by adding two branches, Sales and Support (see [Configuring the Sales Branch](#) and [Configuring the Sales Branch](#)), and define the criteria for the routing.

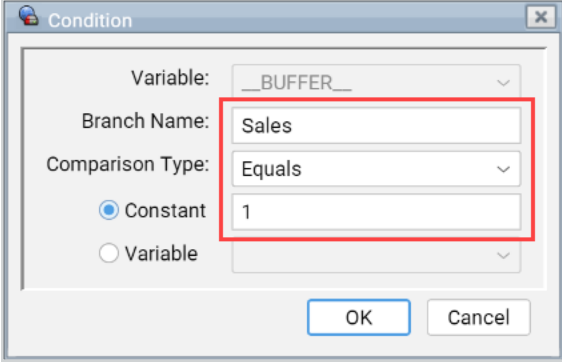
- 2 To configure the Case module, open its properties.
- 3 In the **Variable** field, select the **BUFFER** variable which contains the caller's input.

The screenshot shows the "Case Module Properties" dialog box. The "General" tab is selected. The "Module Name" is "Case2". The "Variable" field is set to "__BUFFER__". Below this is a table with three columns: "Comparison Type", "Value/Variable", and "Branch Name".

Comparison Type	Value/Variable	Branch Name
Equals	"1"	Sales
Equals	"2"	Support
		No Match

Buttons at the bottom: Add, Remove, Edit, Up, Down, Save, Cancel.

- 4 To add a new branch, select **Add**.
- 5 Enter **Sales** in the **Branch Name** field, select **Equals** from the **Comparison Type** drop-down list, and define the **Constant** as 1.

A dialog box titled "Condition" with a close button (X) in the top right corner. It contains the following fields: "Variable:" with a dropdown menu showing "__BUFFER__"; "Branch Name:" with a text box containing "Sales"; "Comparison Type:" with a dropdown menu showing "Equals"; and a section with two radio buttons, "Constant" (selected) and "Variable". Below the "Constant" radio button is a text box containing the number "1". At the bottom are "OK" and "Cancel" buttons. A red rectangle highlights the "Branch Name", "Comparison Type", and "Constant" fields.

Condition

Variable: __BUFFER__

Branch Name: Sales

Comparison Type: Equals

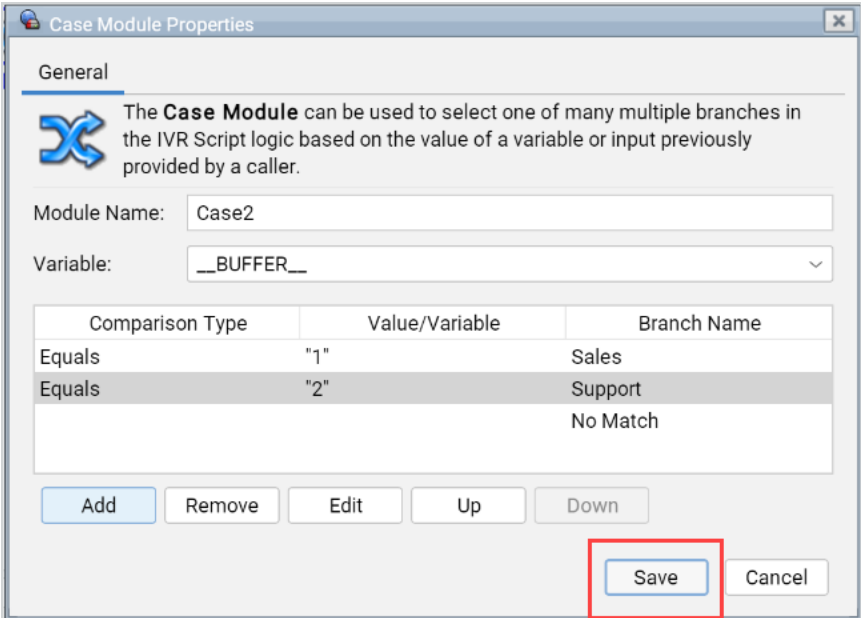
☒ Constant

☐ Variable

1

OK Cancel

- 6 Select **OK**.
- 7 Repeat these steps to create a Support branch, but enter 2 in the **Constant** field.
- 8 Select **Save**.

A dialog box titled "Case Module Properties" with a close button (X) in the top right corner. It has a "General" tab. Below the tab is a blue icon of two crossed arrows and a text description: "The **Case Module** can be used to select one of many multiple branches in the IVR Script logic based on the value of a variable or input previously provided by a caller." Below this are "Module Name:" with a text box containing "Case2" and "Variable:" with a dropdown menu showing "__BUFFER__". Below these is a table with three columns: "Comparison Type", "Value/Variable", and "Branch Name". The table has three rows: "Equals" with "1" and "Sales", "Equals" with "2" and "Support", and an empty row with "No Match". Below the table are buttons: "Add", "Remove", "Edit", "Up", "Down", "Save", and "Cancel". A red rectangle highlights the "Save" button.

Case Module Properties

General

The **Case Module** can be used to select one of many multiple branches in the IVR Script logic based on the value of a variable or input previously provided by a caller.

Module Name: Case2

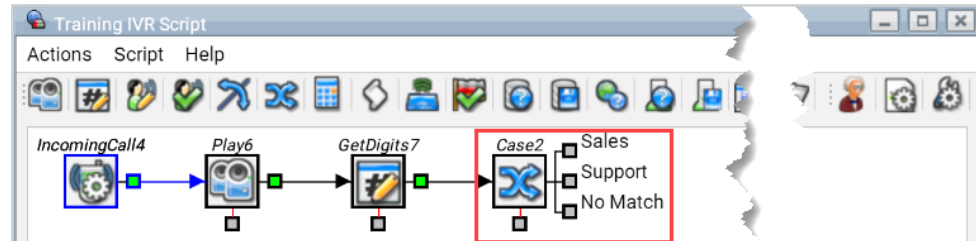
Variable: __BUFFER__

Comparison Type	Value/Variable	Branch Name
Equals	"1"	Sales
Equals	"2"	Support
		No Match

Add Remove Edit Up Down

Save Cancel

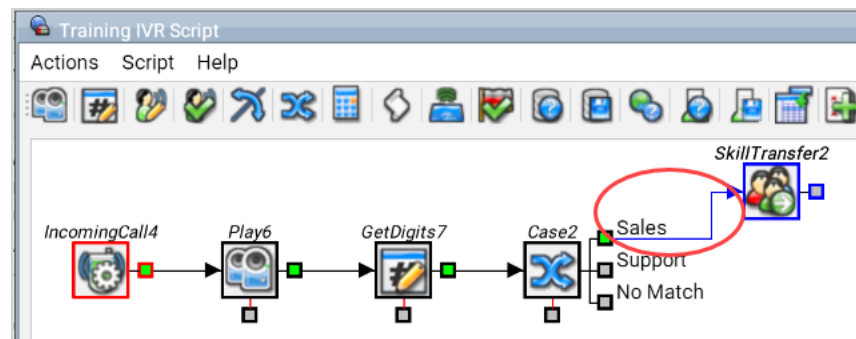
The script flow has three branches: Sales, Support, and No Match.



For our example, we now add the two branches, Sales and Support, and define the criteria for the routing.

Configuring the Sales Branch

- 1 Add the Skill Transfer module, and add a connector to the Skill Transfer module from the Sales branch of the Case module.



Configure the Skill Transfer module to route the call to the appropriate skill group. In this example, the settings of the Skill Transfer module are the same for the Support and Sales branches.

- 2 To configure the Case module, open its properties.

The screenshot shows the 'Skill Transfer Module Properties' dialog box with the 'General' tab selected. The 'Module Name' is 'SkillTransfer2'. Under 'Call Queuing', 'Max Queue Time (mmm:ss)' is set to '010:00' for 'If there are available agents in queue'. The checkbox 'Queue calls when agents in queue, but unavailable (on call)' is checked, also set to '010:00'. The checkbox 'Queue calls when no agents in queue (not ready / logged out)' is unchecked, also set to '010:00'. 'Termination Digit' is 'N/A'. Under 'Queue Callback', 'On queue timeout expiration' and 'On digit receiving' are unchecked. 'Record caller name for agent assigned to QCB' is unchecked. Under 'Transfer to Agent', 'Distribution Algorithm' is 'Longest Wait - All Calls', 'Time Frame' is '15 Minutes', 'Max Ring Time (secs)' is '15', and 'Pause Before Transfer (secs)' is '0'. The checkbox 'Set agent to Not Ready if no answer' is checked. Under 'Transfer to Voicemail', 'On queue timeout expiration' and 'On digit receiving' are unchecked. 'Voicemail Box' is set to 'Skill'. 'Action for Recorded Files' is 'Keep as Recording'. The 'Clear User Input' checkbox is checked. 'Save' and 'Cancel' buttons are at the bottom right.

3 In the **General** tab, set these options:

- **Call Queuing** defines the queue properties of the call:
 - **Max queue time if there are available agents in queue:** Sets the maximum time that should pass from the moment the call transfers to this module, to the moment the call transfers to the next IVR module in flow or to voicemail if agents do not answer during the defined time.
 - **Queue calls when agents in queue, but unavailable (on call):** Select to set the time where the call is on hold waiting until agents can answer.

- **Queue call when no agents in queue (on break/logged out):**
Select to set the time where the call is on hold waiting until the agents can answer.
- **Termination Digit:** Choose the number the caller presses to pass the call to the next IVR module.
- **Queue Callback Digit:** Choose the number the caller presses to receive a callback when their position in the queue arrives.
- **Transfer to Agent** defines the call routing for the agent's transfer process:
 - Distribution algorithm specifies the way the ACD selects an agent to receive a call. The configured distribution algorithms are available from the drop-down list. Select an algorithm that meets the requirements of your enterprise:
 - **Longest Wait - All Calls:** The ACD selects the agent with the longest wait (idle) time since their last call, of any call type.
 - **Longest Wait - Non-Manual:** The ACD selects the agent with the longest wait (idle) time excluding manual calls.
 - **Round Robin:** The ACD selects an agent from the list of logged-in agents who qualify for the queue. The list is initially sorted based on the longest wait (idle) time for all calls. When agents initially make themselves ready for the queue, they are added to the bottom of the list. The first queued call goes to the agent on top of the list. When an agent receives a call, the agent is removed from the list, until all agents are removed. For queued calls, when the only available agents are those already removed from the list, the calls are still routed to those agents.
 - **Min Calls Handled:** The ACD prioritizes agents who have handled fewer calls than the other agents working in the queue, based on statistics. The number of calls is calculated from the agent's last login (that is, the *current session*), and then assessed for the selected time range (see the Time Frame option below).

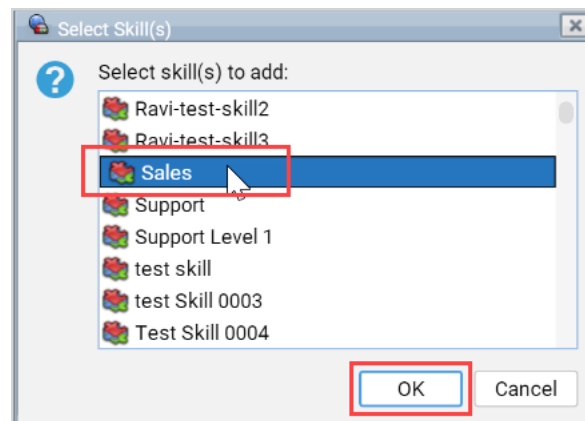
Note: The time range counter for Min Calls Handled and Min Handle Time begins at agent login and resets to zero (0) at the interval set in the Time Frame option in the [Skill Transfer Module](#).

- **Min Handle Time:** The ACD prioritizes agents with the lowest total handle time compared to the other agents working in the queue, based on statistics. The total handle time is calculated from the agent's last login (that is, the *current session*), and then assessed for the selected time range (see the Time Frame option below).
- **Time Frame:** Time range for the Min Calls Handled or Min Handle Time distribution algorithms.
- **Max Ring Time (secs):** The maximum time for the agent to answer a call. The option allows you to set timeout in seconds.
- **Pause Before Transfer (secs):** Creates a delay before sending the call to the agent. When the agent is selected for a new call, the system waits the set amount of time before sending the call to the agent. When there are queued calls, this option gives agents a short pause before receiving the next call.
- **Set Agent to Not Ready if No Answer:** Select to automatically change agent status to Not Ready if they don't answer the call within the Max Ring Time.
- **Transfer to voicemail** contains the options that define the call transfer to voicemail.
 - **On queue timeout expiration:** Transfer calls to the skill group voicemail when the time defined in the queue area options expires.
 - **On digit receiving:** Number caller presses to transfer the call to the skill group voicemail.
 - **Voicemail Box** (skill or personal voicemail box): Selects the skill group or agent to receive the voice message from the caller.
- Action for recorded files allows you to define the action to take for recorded files gathered by the previous modules. See also [Using Recorded Files](#).
 - **Clear User Input:** Clears the user input when the call enters this module.

4 Select the **Skills** tab.

In our example, configure one Skill Transfer module to use the Sales skill group and the second to use the Support skill group. In the Skills window, add the skill groups that receive the calls and define the sequence to use to choose the skill group for transferring the call. The upper group receives the call first if there are agents available. Otherwise, the call transfers to the next skill group in the list with available agents.

- 5 Select **Add**.
- 6 Select the Sales skill group, and click **OK**.

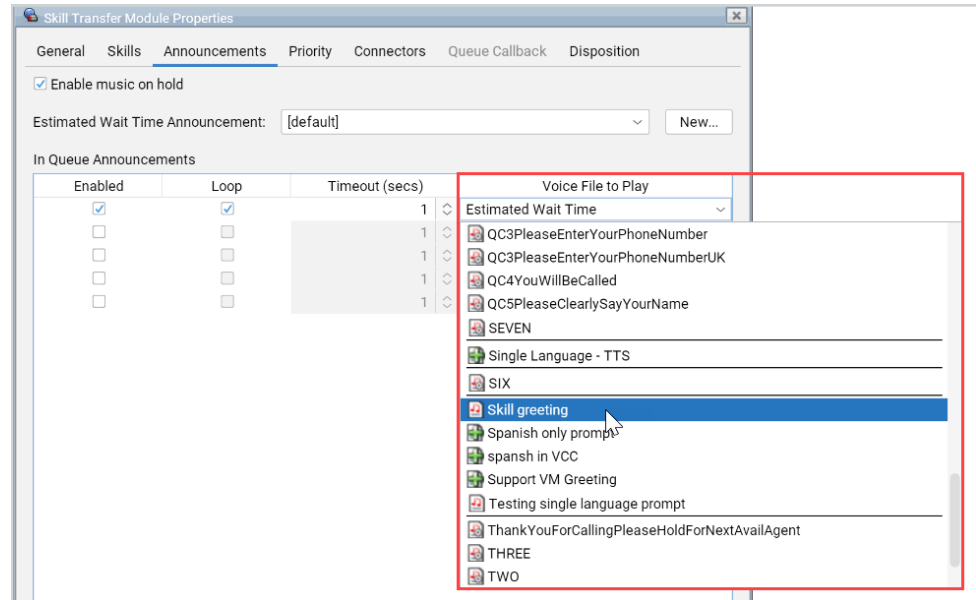


The skill group appears in the list. Use the + and - buttons to change the skill position in the list.

- 7 Select the **Announcements** tab.

Set the voice messages that play to the caller while the call is on hold in queue. The **Enable music on hold** option plays the default hold music to the caller while the call is in the queue. You can set up to five announcement messages to play to the caller at defined times during their wait time.

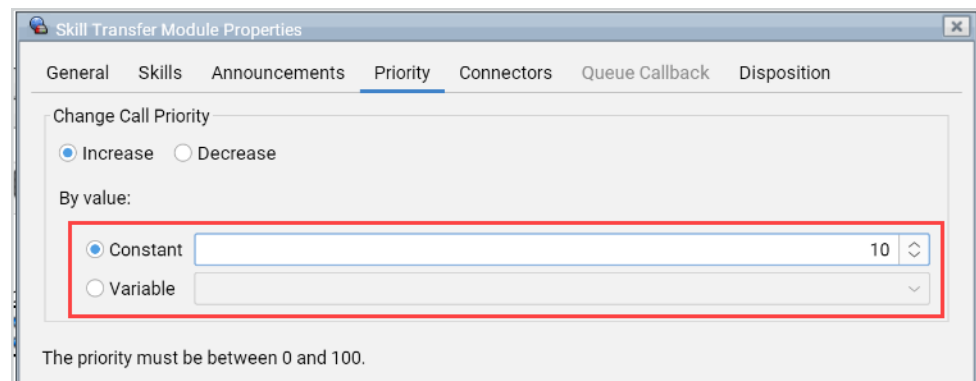
- a To add an announcement, select the **Enabled** option and set a time in the **Timeout** column.
- b To repeat the message through the period of time defined in the Timeout column, select the **Loop** option.
- c To add the voice message to play to callers, select the message in the **Voice File to Play** column and select the message from the list. You can add voice messages using the **Add Prompt** command in the Prompts folder. See [Adding Prompts](#).



8 Select the **Priority** tab.

Use priority management when you use one skill group in different script branches and the call from one branch to initially be accepted by available agents. You can modify the call priority. Priority means how fast the call transfers to agents from the queue. By default, Basic mode campaign calls begin with a call priority of 10 and Advanced mode calls have priority of 60. The priority range is from 0 to 100. You can increase priority to make the call transfer to the agent faster or decrease it to leave the call in queue for more time. For more information about campaign modes, see the [Campaign Administrator's Guide](#).

9 Enter the value to change the priority in the **Constant** field or select a **Variable** integer value.



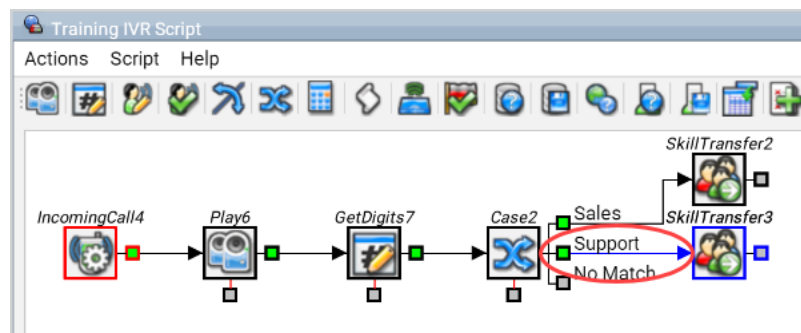
- 10 Select the **Dispositions** tab to define the disposition of the call if the caller hangs up.

The default disposition is **Abandon**.

- 11 Select **Save**.

Configuring the Support Branch

For the Support branch, add the Skill Transfer module and link it to the Support branch. Configure the Skill Transfer module of the Support branch to route the call to the appropriate skill group.

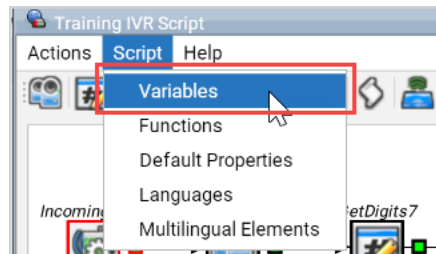


Configuring the No Match Branch

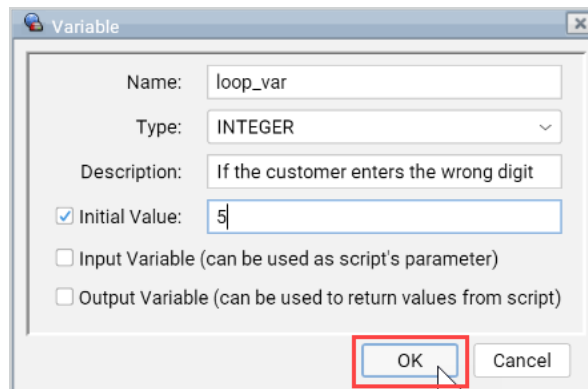
Now that we have the flow of the script, we need to configure the script to handle wrong input from the caller. For this we use the No Match branch of the Case module.

In our example, if caller makes an invalid entry (a number other than 1 or 2), they can repeat the input. After five consecutive invalid entries, the call is disconnected. If the caller's entry is invalid, we repeat the voice message instructions and have the caller try again. To trace the number of invalid entries, create the variable `loop_var` in the Variables tab to contain the number of allowed wrong entries (by default, Initial Value is 5).

- 1 In the Script menu of the IVR Script Properties window, select **Variables**.



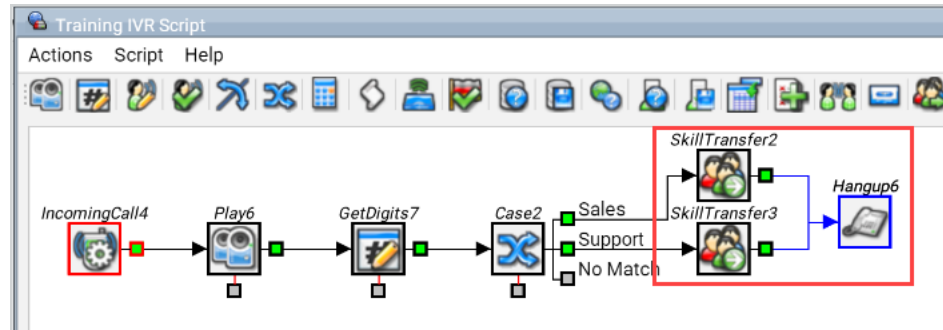
- 2 To add a new variable, select **Add**.
- 3 Enter the name `loop_var`.
- 4 Select the **Type** of variable: **INTEGER**.
- 5 Enter a description for the variable.
If the customer enters the wrong digit, in this example.
- 6 Select the **Initial Value** option and enter 5.



- 7 Select **OK**.

Adding the Hang Up Module

Now that both Skill Transfer modules are configured, we'll add a script termination module. Connect the Hang Up module to both Skill Transfer modules.

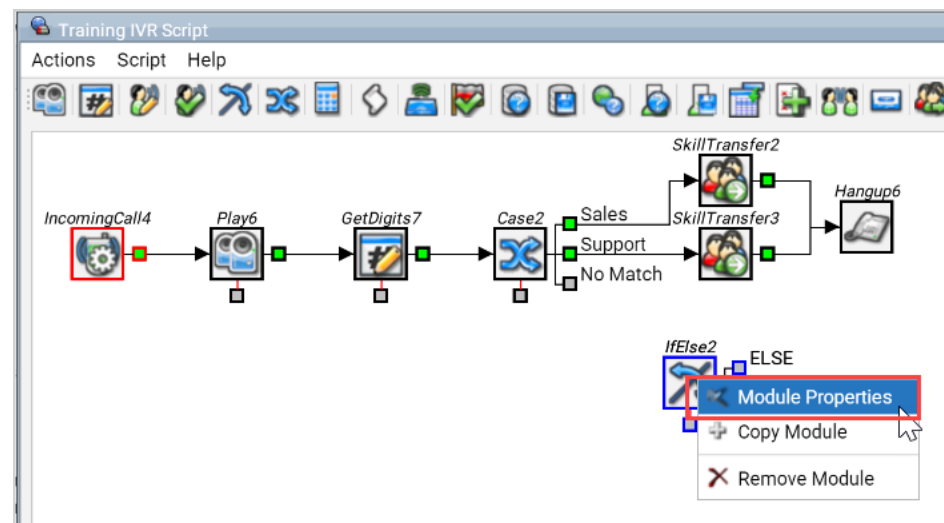


Adding and Configuring the If/Else Module

We configure this module to check the `loop_var` variable that was initialized at the beginning of the script with the 5 initial value. If the value of the variable is less than 1, the caller has entered incorrect digits five times. We should terminate the script.

- 1 Add the If /Else module to the script.
- 2 Right-click **If/Else**.

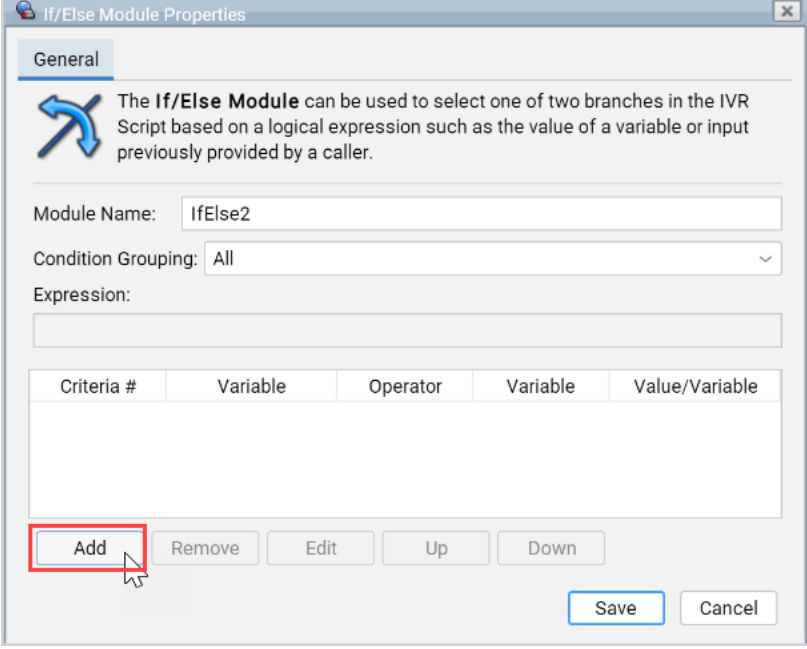
The Module Properties window displays.



Create the logical expression checked during the script.

If the result of expression is true, the script uses the If branch. If false, the script uses the Else branch.

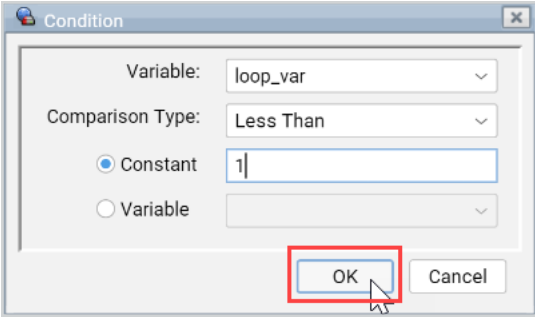
- 3 Click **Add** to add the condition to check.



The If/Else Module Properties dialog box is shown. The 'General' tab is selected. It contains a description of the module, a 'Module Name' field set to 'IfElse2', a 'Condition Grouping' dropdown set to 'All', and an 'Expression' field. Below these is a table with columns: Criteria #, Variable, Operator, Variable, and Value/Variable. At the bottom, there are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', 'Save', and 'Cancel'. The 'Add' button is highlighted with a red box and a mouse cursor.

Criteria #	Variable	Operator	Variable	Value/Variable
------------	----------	----------	----------	----------------

- 4 From the **Variable** drop-down list, select the variable to check: `loop_var`.
- 5 Select the comparison type for the condition from the **Comparison Type** drop-down list: Less Than.
- 6 In the **Constant** field, enter the value to compare the variable with the comparison type: 1.



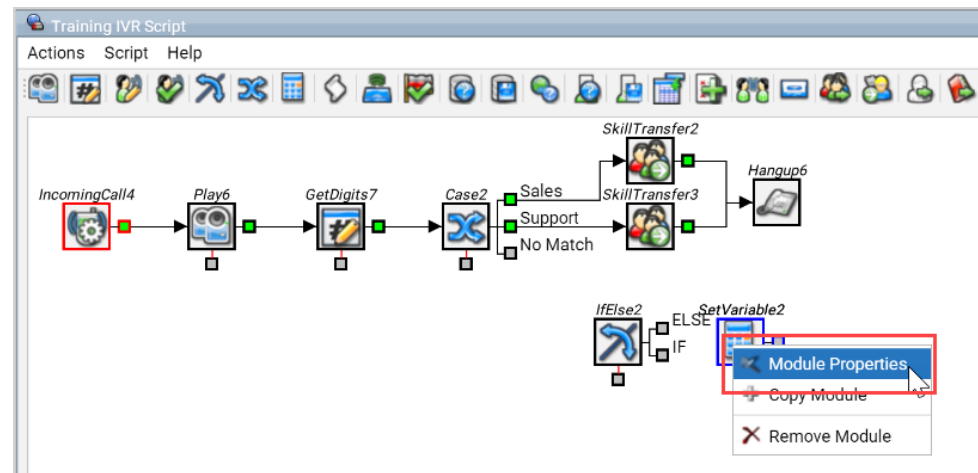
The Condition dialog box is shown. It has fields for 'Variable' (set to 'loop_var') and 'Comparison Type' (set to 'Less Than'). There are two radio buttons: 'Constant' (selected) and 'Variable'. The 'Constant' field contains the value '1'. At the bottom, there are 'OK' and 'Cancel' buttons. The 'OK' button is highlighted with a red box and a mouse cursor.

- 7 Click **OK**.
- 8 Click **Save**.

Adding and Configuring the Set Variable Module

We configure this module to subtract 1 from `loop_var` each time the call enters this module. If the caller enters incorrect information five times, the `loop_var` variable contains 0 after the Set Variable module completes.

- 1 Add the Set Variable module to the script.



- 2 Open the Set Variable module properties window.
- 3 Click **Add** to add the variable to assign to this module.

The Assignment dialog box shows the following configuration:

- Variable: `loop_var`
- Type: `INTEGER`
- Description: `If the customer enters the wrong dig`
- Assigned Value: `5` (Constant selected)

The OK button is highlighted.

- 4 From the **Variable** drop-down list, select the variable: `loop_var`.
- 5 Select the **Function Assigned Value** option.
- 6 Select the **DIFFERENCE(INTEGER,INTEGER)** function from the drop-down list.
- 7 Define the function arguments in the corresponding rows.

The first argument is the variable named `loop_var`.

To define this:

- a Click the first column in the #1 row and select **Variable**
- b Click the second column and choose `loop_var`.
- c Define the second argument.

In our example, this is the constant with the value equal to 1.

Assignment

Variable

loop_var

Type: INTEGER

Description: If the customer enters the wrong dig

Assigned Value

☐ Constant ☐ Variable ☒ Function ☐ NULL

DIFFERENCE(INTEGER, INTEGER)

Function Arguments:

#1	Constant	loop_var
#2	Constant	1

OK Cancel

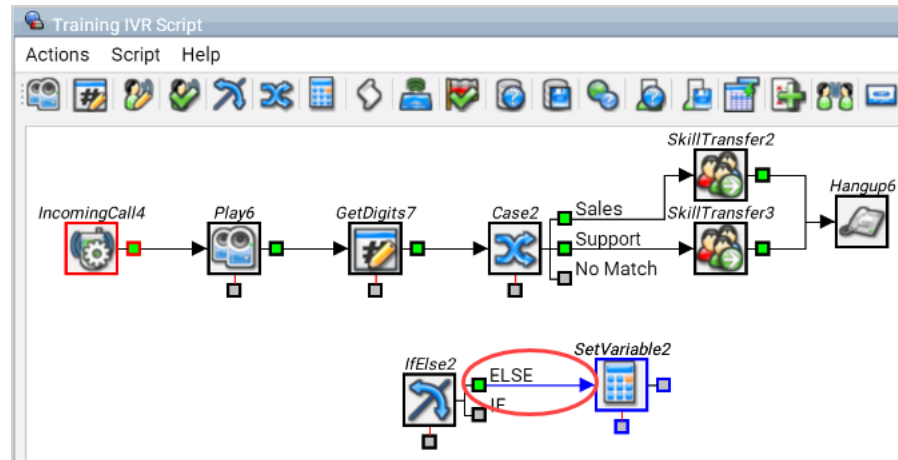
- 8 Select **OK**.
- 9 Select **Save**.

Linking From the If/Else Set Variables Modules

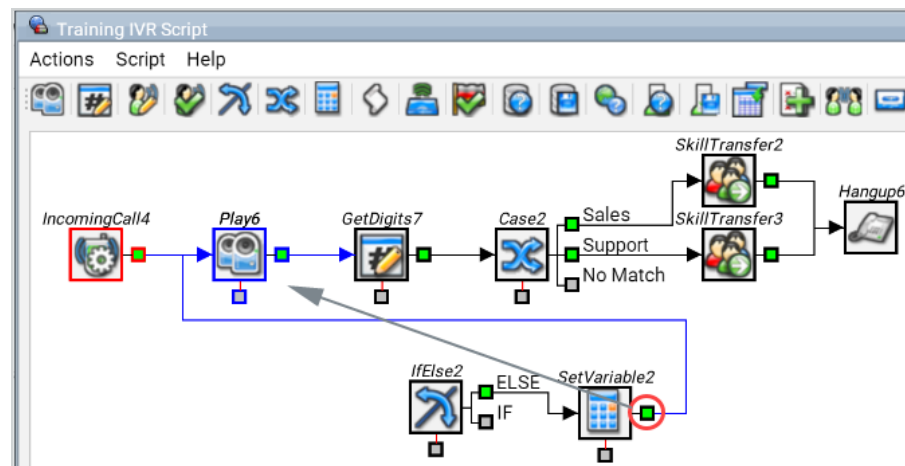
If the caller enters the wrong digit five times, the script follows the If branch of the If/Else module, which means that you should connect the If branch to the Hang Up module.

If the caller has entered the wrong digit fewer than five times, the call is returned to the Play module at the beginning of the script after reducing the number of available wrong entries in the Set Variables module.

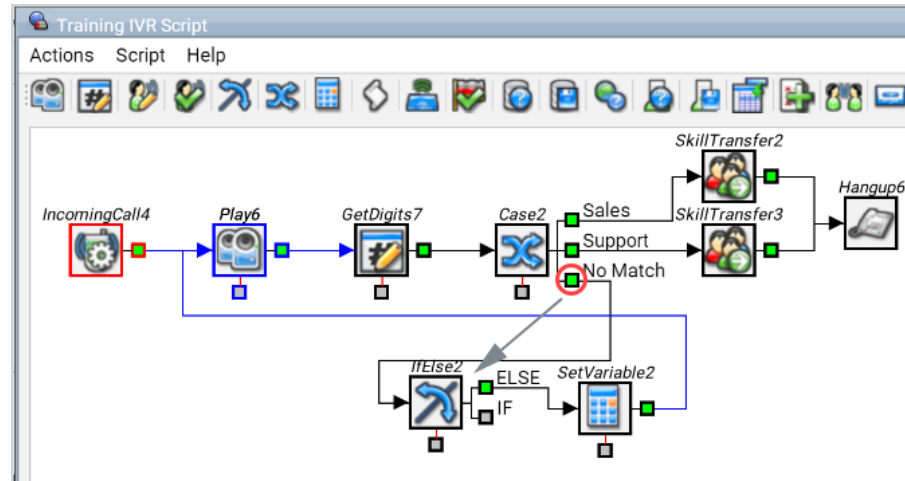
- 1 Link the Else branch of the If/Else module to the Set Variables module.



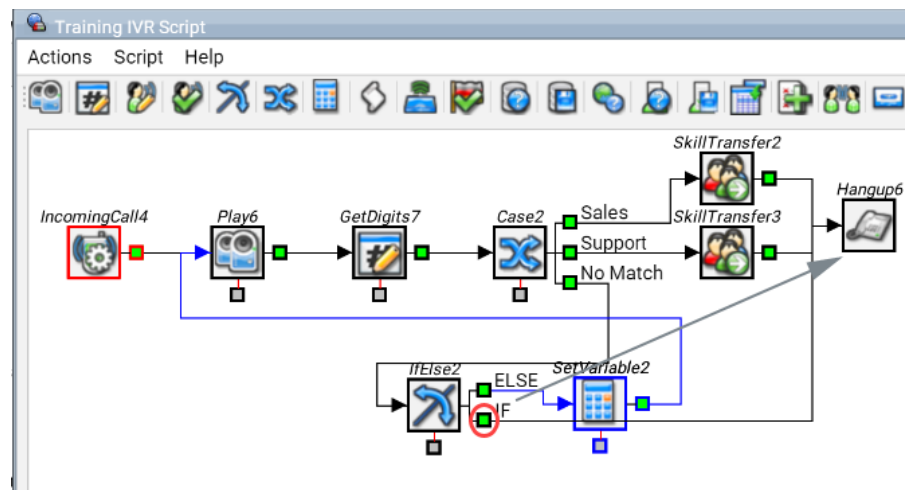
- 2 Link the Set Variables module to the Play module.



- 3 Connect the No Match branch with the If/Else module.



- 4 Connect the If branch with the Hang Up module.



- 5 To save the script, select **Actions > Save**.

You can download this example to import it into your environment.

Creating an IVR Loop

This example shows how to create a loop, use variables, and use the Set Variable module. We loop from the No Match branch of the Case module when a caller makes an invalid entry.

There are other methods for creating loops. This is just one example. Another option is to set the loop variable to zero, then use If/Else to compare for the maximum number of loops and use SetVariable to add to, rather than subtract from, the loop variable. You can also insert a Play module inside the loop to inform the caller that they made an invalid entry.

[Defining a Variable](#)

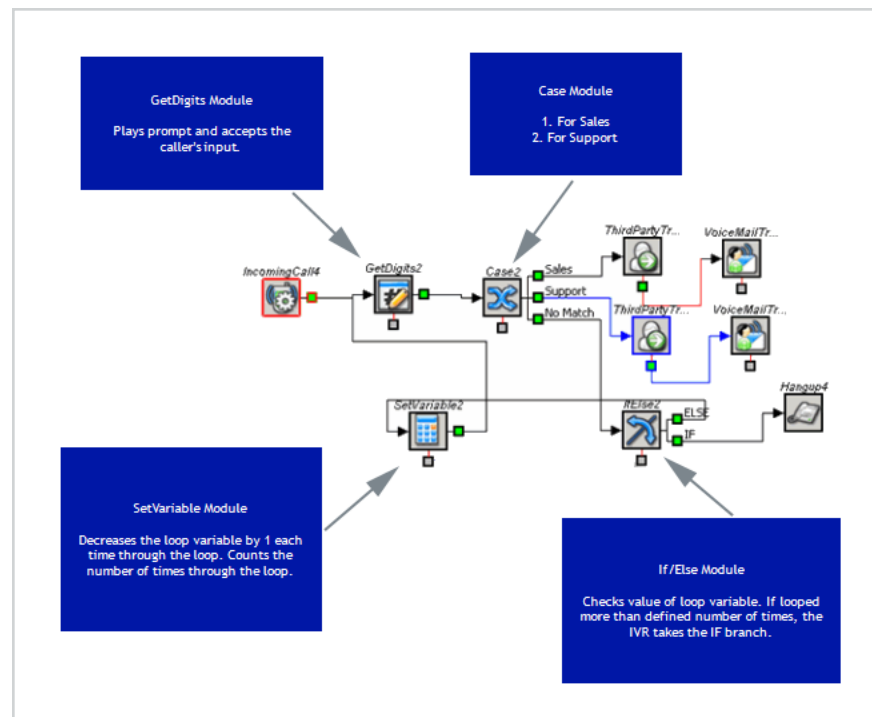
[Adding and Configuring the Get Digits Module](#)

[Adding and Configuring the Case Module](#)

[Adding and Configuring the If/Else Module](#)

[Adding and Configuring the Set Variable Module](#)

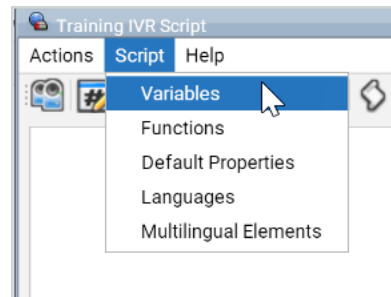
[Linking the Modules](#)



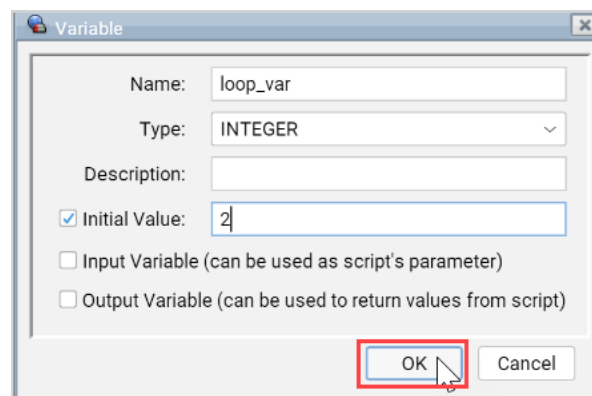
Defining a Variable

The first step in creating a loop is to define a variable. Set the initial value of the variable to the number of times you want to loop. For each loop we subtract one from this number. If the variable reaches zero, the loop will end.

- 1 In the Visual IVR Script designer, select Variables from the Script menu.



- 2 Click **Add**.
- 3 Enter a name for the variable - in our case the name is `loop_var`.
- 4 Select INTEGER as the Type (an integer is any whole number, positive or negative). Since we will be doing a calculation on this variable, subtracting 1 for each loop, it must be integer or numeric rather than a string.
- 5 In the Initial Value box, enter the maximum number of times you want to loop: 2.

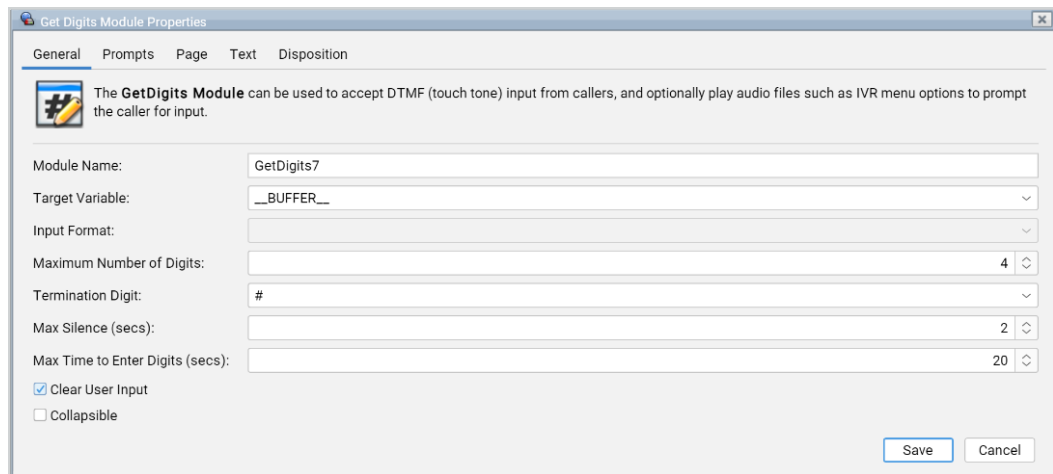


- 6 Click **OK** to save the variable.
- 7 Click **OK** in the Variables window.

Adding and Configuring the Get Digits Module

Configure the Get Digits module to accept an entry from the caller. In our example, we accept a single digit (either 1 or 2) so we set the Maximum Number of Digits to 1.

We are using the default BUFFER system variable. The caller's input is stored in this variable.



The screenshot shows the 'Get Digits Module Properties' dialog box with the 'General' tab selected. The dialog contains the following fields and options:

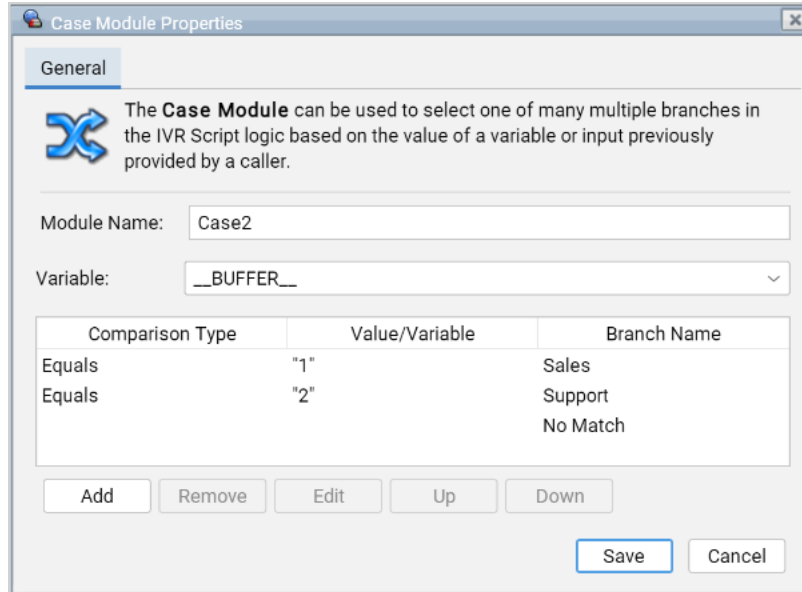
- Module Name:** GetDigits7
- Target Variable:** __BUFFER__
- Input Format:** (empty)
- Maximum Number of Digits:** 4
- Termination Digit:** #
- Max Silence (secs):** 2
- Max Time to Enter Digits (secs):** 20
- ☒ Clear User Input
- ☐ Collapsible

Buttons for 'Save' and 'Cancel' are located at the bottom right.

Adding and Configuring the Case Module

Configure the Case module with branches based on the caller's input. We are comparing the value stored in the BUFFER system variable which was set by the caller's input in the Get Digits module. Our loop will be used for the No Match branch.

For looping we use the combination of the If/Else module and the Set Variable module to determine how many times a caller has cycled through the loop. Set Variable decreases the loop variable by 1 each time through the loop. If/Else checks the value of the loop variable to determine if the caller has looped the maximum number of time.



The Case Module Properties dialog box is shown with the General tab selected. It contains a description of the Case Module, a Module Name field set to 'Case2', and a Variable dropdown menu set to '__BUFFER__'. Below these is a table with three columns: Comparison Type, Value/Variable, and Branch Name. The table contains two rows: 'Equals' with value '1' leading to 'Sales', and 'Equals' with value '2' leading to 'Support'. A 'No Match' branch is also listed. At the bottom are buttons for Add, Remove, Edit, Up, Down, Save, and Cancel.

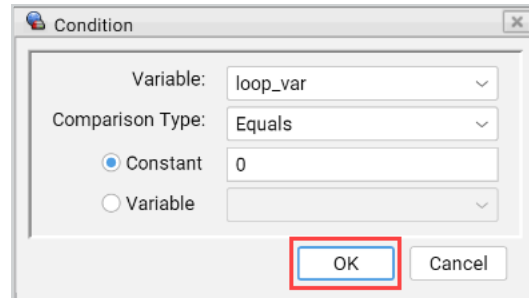
Comparison Type	Value/Variable	Branch Name
Equals	"1"	Sales
Equals	"2"	Support
		No Match

Adding and Configuring the If/Else Module

If a caller makes an invalid entry or no entry at all, the IVR use the No Match branch. The first module in the No Match branch is used to check the value of the loop variable. We use the If/Else module to do the comparison.

If the value is equal to zero, we use the If branch. If it is greater than zero we use the Else branch. The Else branch is the loop. Since the initial value of the loop variable is 2 the first time through, the If/Else module uses the Else branch.

- 1 In the If/Else module click **Add** to add a new comparison.
- 2 In the Variable list select the loop variable you created (`loop_var` in this case).
- 3 In the Comparison Type list, select Equals.
- 4 In the Constant box enter 0 (zero).



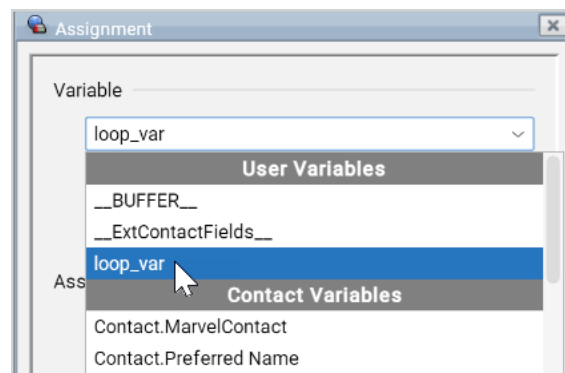
- 5 Click **OK**.

This module checks if the value of the loop variable is zero.

Adding and Configuring the Set Variable Module

After the If/Else module, we use the Set Variable module to decrease the loop variable by 1. This module is basically counting the number of times the caller cycles through the loop. In our example we are counting down from the maximum number of loops set by the initial value of the loop variable (`loop_var = 2`).

- 1 In the Set Variable module, click **Add** to add a new entry.
- 2 Select the loop variable from the Variable drop-down list.



- 3 Select Function for the Assigned Value.

The 'Assignment' dialog box is shown with the following settings:

- Variable:** loop_var
- Type:** INTEGER
- Description:** If the customer enters the wrong dig
- Assigned Value:** The 'Function' radio button is selected and circled in red.
- Function Arguments:** An empty text area.

Buttons at the bottom: OK, Cancel.

- 4 Select DIFFERENCE(INTEGER, INTEGER) in the drop-down list. We will use this function to subtract 1 from the loop variable.

The 'Assignment' dialog box is shown with the 'Function' dropdown menu open. The list of functions includes:

- DIFFERENCE(INTEGER, INTEGER) (highlighted by a mouse cursor)
- DIFFERENCE(DATE, DATE)
- DIFFERENCE(TIME, TIME)
- LEN(String)
- LOCATE(String, String)
- PRODUCT(INTEGER, INTEGER)
- QUOTIENT(INTEGER, INTEGER)
- SIZE(KVList)

- 5 In the #1 Function Argument, click the second column and select Variable.

Assignment

Variable

loop_var

Type: INTEGER

Description: If the customer enters the wrong dig

Assigned Value

☐ Constant ☐ Variable ☒ Function ☐ NULL

DIFFERENCE(INTEGER, INTEGER)

Function Arguments:

#1	Constant	
#2	Constant	

OK Cancel

6 In the third column for #1, select the loop variable (`loop_var` in our example).

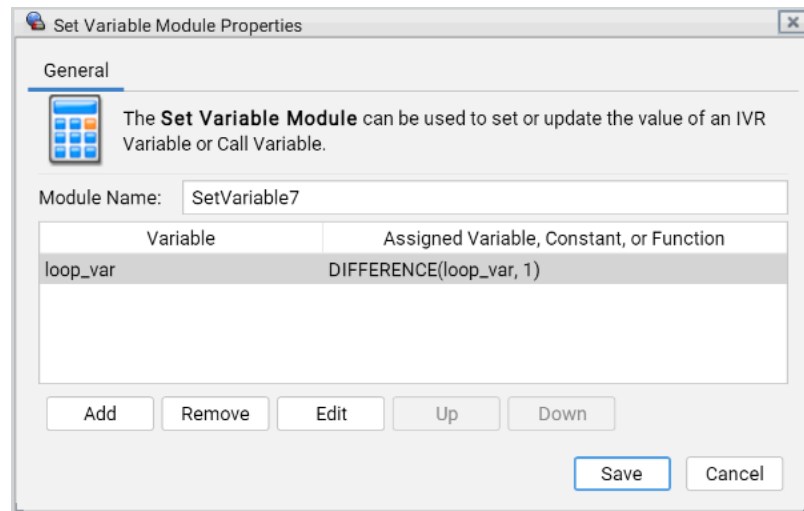
7 In the #2 Function Argument, select Constant in the second column.

8 In the third column for #2, enter 1.

We subtract this value from the loop variable.

9 Click **OK**.

The assignment is added to the Set Variable module.



Linking the Modules

- 1 Link the If/Else module from the No Match branch of the Case module.
- 2 Link the If branch of the If/Else module to the IVR branch used when the maximum number of loops is reached.
- 3 Link the Else branch of the If/Else module to the Set Variable module.
- 4 Link the Set Variable module to the Get Digits module.

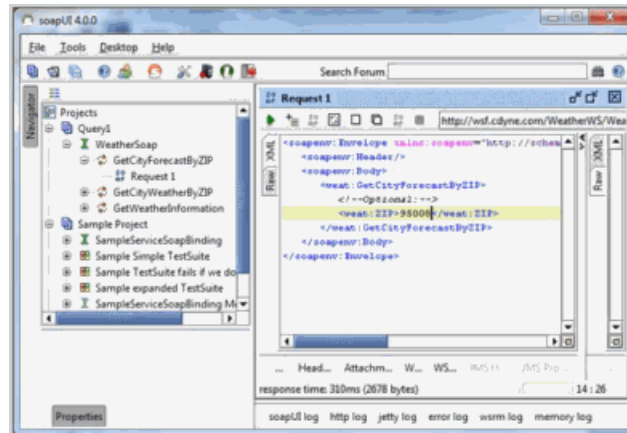
This returns the caller back to the original prompt. See the first screen shot for an example.

Configuring the Query Module

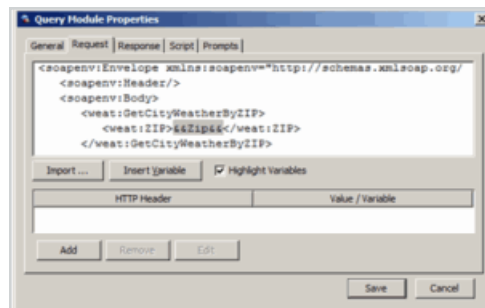
The Query module supports XML-encoded POST requests and replies. The module invokes complex Web Services, for example, Web Services based on SOAP such as Five9 Web services. You can use external tools to confirm and debug the Web Service before entering the request and response templates into the Query module. In our case an example of using SOAP UI is shown.

Use the Iterator Module to update variables in the Query Module with next matches result set.

- 1 With SOAP UI, form a proper request for the Web Service.

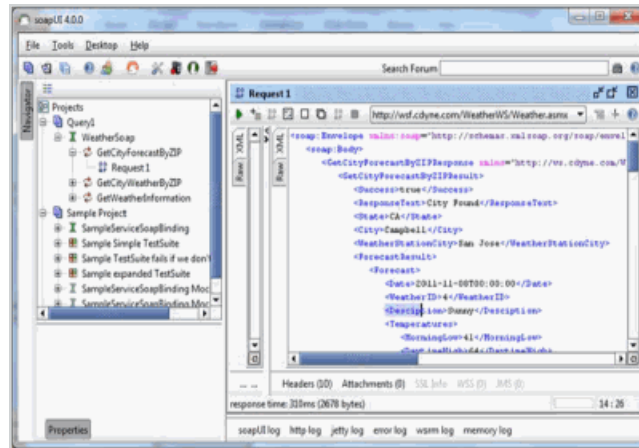


- 2 Paste the XML code into the Request tab of the Query Module and embed variables from IVR.

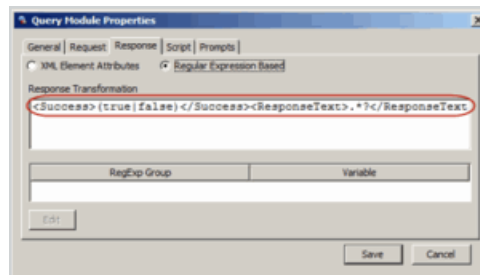


The Insert Variable button allows you to choose input variables and delimits them. In our example, it is &&Zip&&.

- 3 Call the Web Service from SOAP UI to get an example Response.



- 4 Copy the response and paste it into the Regular Expression tool.

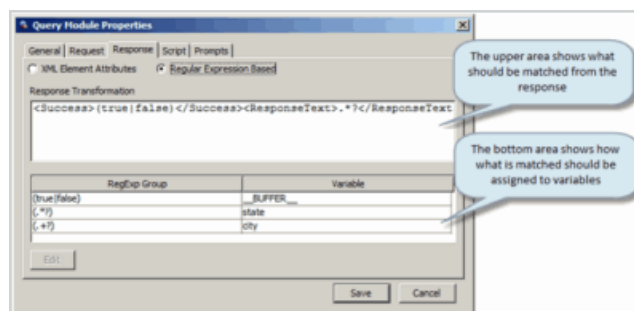


- 5 Insert Regular Expressions where you want to match part of the Response.

To find the best regular expression, you can use a tool such as this:

<http://gskinner.com/RegExr/?31e98>.

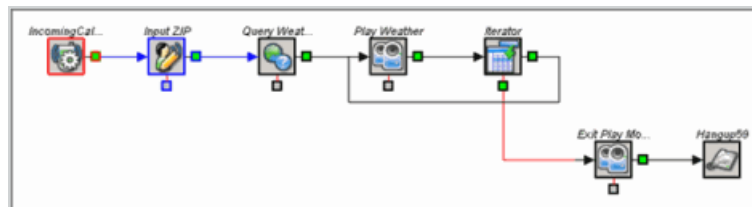
- Enter the Response from your Web Service.
- Enter your pattern to match.
- Use parenthesis around expressions which should be matched and assigned to variables.
- Test using evaluator.



- 6 Once the Regex tool is matching the data you want, assign the regular expressions to variables in the Response tab (bottom area).
- 7 Run the IVR Script to test.
- 8 When testing is successful, enter the pattern into the Response tab and run the IVR script with the Query Module.

Creating an IVR Script with XML Web Service Query

This topic provides information on how to configure original campaign's IVR script with XML Web Service Query. This example gets weather forecast by sending a query with a ZIP code to a Web service. We use script variables but not call variables because the caller is not connected to an agent.



Input Module

Using the Input Module the caller enters a ZIP code. In the General tab of the Input Module Properties, choose ZIP as a variable.

The **Input Module Properties** dialog box is shown with the **General** tab selected. It contains the following fields and options:

- Module Name:** Input ZIP
- Variable:** Zip
- Grammar:** zipcode
- Properties:** A table with columns **Appl**, **Property**, and **Value**.

Appl	Property	Value
☒	language	en-US
- Additional Returned Values:** A table with columns **Return Key** and **Variable to Save Value**.

Return Key	Variable to Save Value
SWL_literal	<ignore>
- Max Attempts:** 3
- DTMF Help Buttons:** <none>
- ☐ Record User's Input as Audio File

Buttons: **Save**, **Cancel**

Query Module

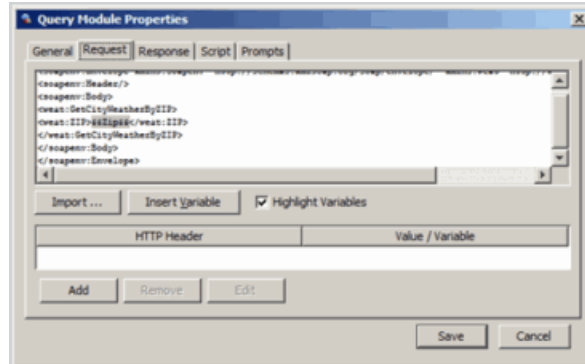
- 1 In the General tab of the Query Module Properties window, enter the URL of the SOAP Web service into the URL field and choose XML Web Service POST.

The **Query Module Properties** dialog box is shown with the **General** tab selected. It contains the following fields and options:

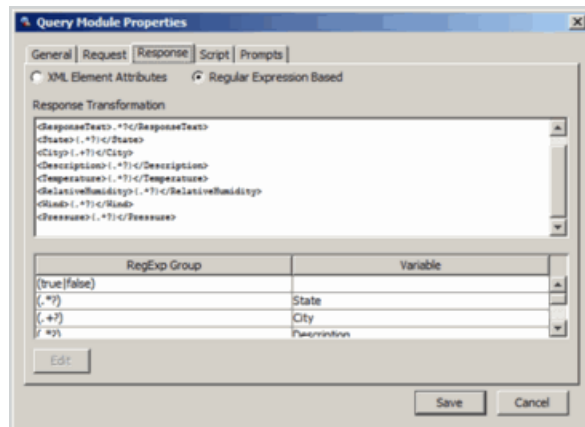
- Module Name:** Query Weather
- URL:** http://wsf.cdyne.com/WeatherWS/Weather.asmx
- Method:** XML Web Service POST
- Fetch timeout (secs):** 7
- ☐ Save Number of Records to:

Buttons: **Save**, **Cancel**

- 2 In the Request tab, enter the SOAP request.
It should contain the ZIP variable delimited by &&. This request will be sent to the weather service.

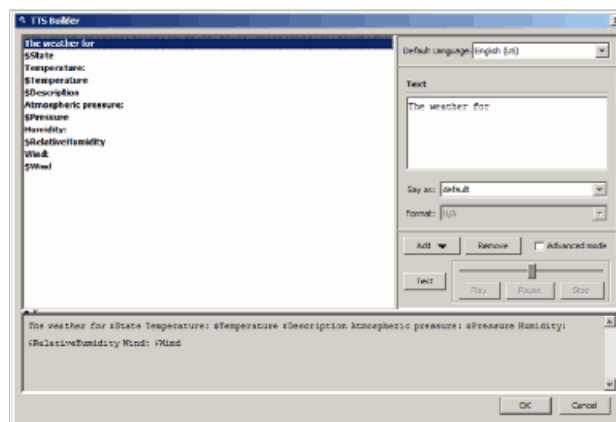
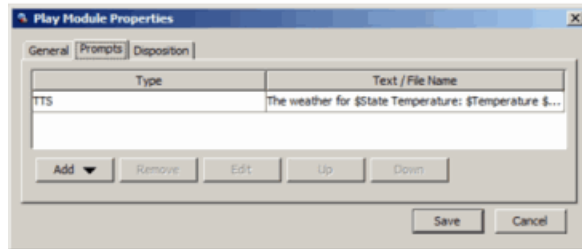


The Response Transformation area allows an input of Embedded Regular Expressions that can be mapped to IVR variables in the RegExp Group area. Any of regular expressions that are grouped (are enclosed within parenthesis), in the Response Transformation section, are automatically loaded in the table below as the RegExp Group. You should assign these regular expressions to variables.



Play Module

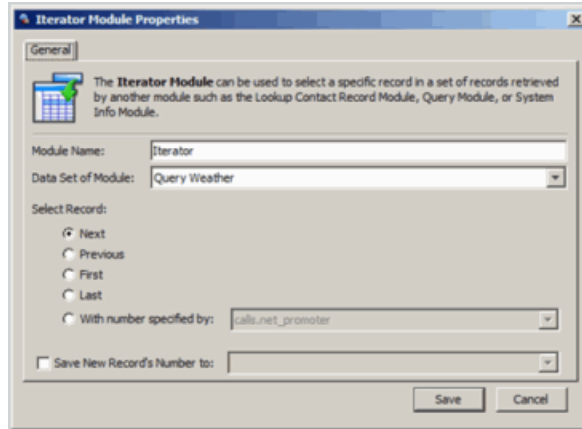
The Play Module plays the response of the web service to the caller. In the Prompts tab, add TTS prompt to be played to the user. The text of the prompt contains the information received from the Web service.



Iterator Module

The Iterator Module updates variables if you have multiple matches from the regular expressions. Select the Query Module that we configured earlier in the Data Set of Module list and chose **Next**.

Each time the Iterator Module is triggered, the Play Module moves to the next variable and the caller hears the further information.



Creating an IVR Script with Queue Callback

This section describes how to configure IVR script with queue callback for a campaign.

In the agent's desktop, the ACD Status tab shows the call when it is in queue. Once the agent becomes ready, the station automatically dials the phone number indicated by the caller. During/after-call - variables are displayed.

There a special disposition for Queue Callback calls. Based on this disposition, reports can be generated.

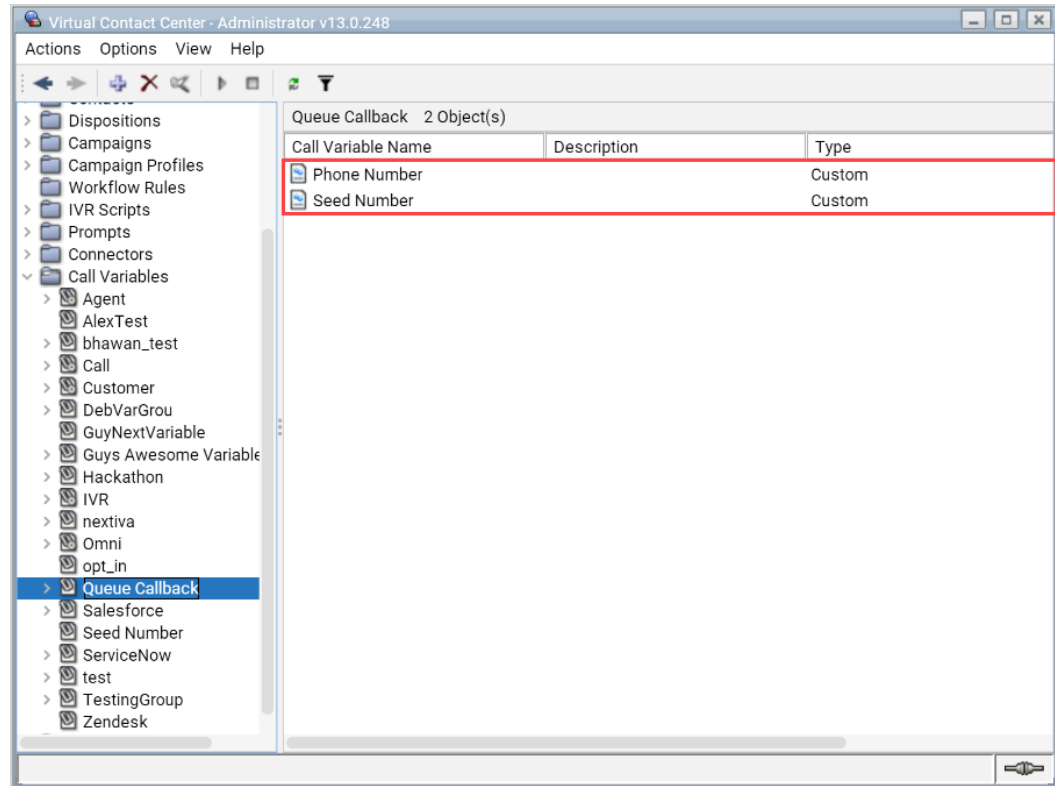
Call Variables Configuration

To create a Queue Callback Variable group, right-click the Call Variables folder and select **Add Call Variable Group**.

Create custom variables for questions in the script example:

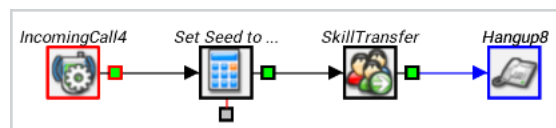
- Seed Number - variable (call type) used for the initial caller's number.
- Phone Number - variable (call type) used for phone number indicated by the user.

Read more in Creating Call Variable Groups, Creating Call Variables.



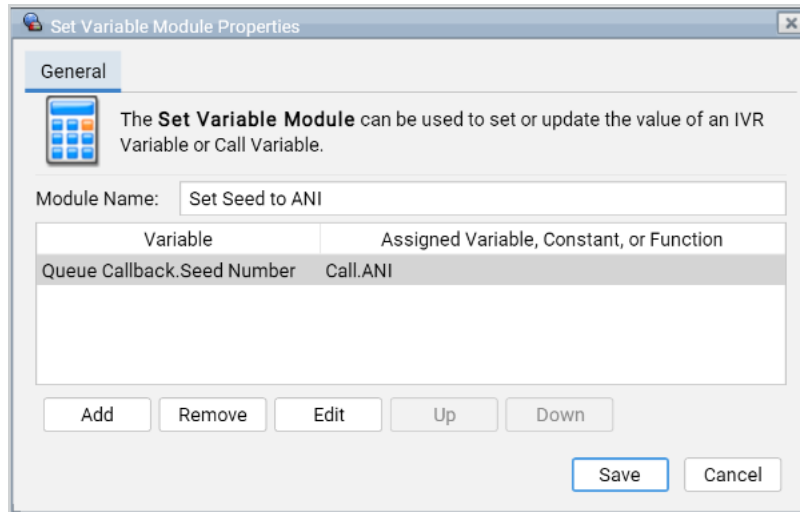
IVR Script Configuration

Let's see how the Queue Callback script is configured.



Set Variable Module

The Set Variable module is used to assign ANI to the Queue Callback Seed Number variable.



Set Variable Module Properties

General

The **Set Variable Module** can be used to set or update the value of an IVR Variable or Call Variable.

Module Name: Set Seed to ANI

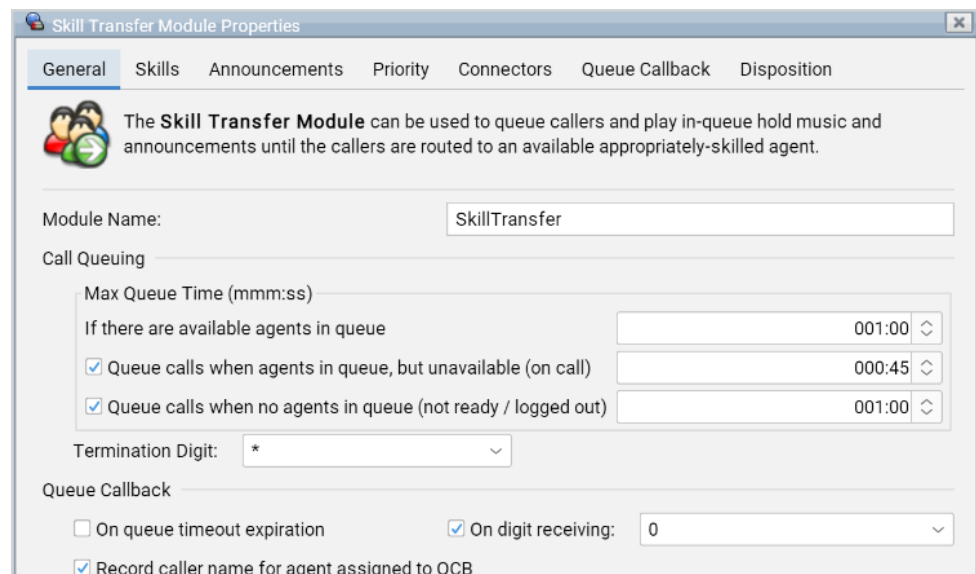
Variable	Assigned Variable, Constant, or Function
Queue Callback.Seed Number	Call.ANI

Add Remove Edit Up Down

Save Cancel

Skill Transfer Module

- 1 In the Queue Callback Digit list, select the Queue Callback Digit other than N/A. In our case, 0 was selected.



Skill Transfer Module Properties

General Skills Announcements Priority Connectors Queue Callback Disposition

The **Skill Transfer Module** can be used to queue callers and play in-queue hold music and announcements until the callers are routed to an available appropriately-skilled agent.

Module Name: SkillTransfer

Call Queuing

Max Queue Time (mmm:ss)

If there are available agents in queue 001:00

☒ Queue calls when agents in queue, but unavailable (on call) 000:45

☒ Queue calls when no agents in queue (not ready / logged out) 001:00

Termination Digit: *

Queue Callback

☐ On queue timeout expiration ☒ On digit receiving: 0

☒ Record caller name for agent assigned to OCB

- 2 In the Queue Callback tab, from the Initial Callback Number from list, select the Queue Callback Seed Number variable.
- 3 From the Store Callback Number in list, select the Queue Callback Phone Number variable.

4 In the Prompts section, choose the prompts.

Example prompt messages:

- Your phone number is ...
- Press one if correct.
- Please enter your phone number.
- You will be called.

5 Indicate the Disable Queue Callback after and After Digit Timeout options.

The screenshot shows the 'Skill Transfer Module Properties' dialog box with the 'Queue Callback' tab selected. The dialog has several sections: 'General' (Initial Callback Number from: Queue Callback.Seed Number, Store Callback Number in: Queue Callback.Phone Number, Allow collection and callback of International numbers? checkbox), 'Prompts' (Playing the Phone Number: QC1YourPhoneNumberIs, Confirming the Phone Number: QC2PressOneIfCorrect, Entering the Phone Number: QC3PleaseEnterYourPhoneNumber, Recording Caller's Name: QC5PleaseClearlySayYourName, Confirmation and Thank You: QC4YouWillBeCalled), 'Timeouts' (Disable Queue Callback after: Days: 1, Hours: 5, Minutes: 0, Seconds: 0; After Digit Timeout (secs): 5), and 'TCPA Consent' (TCPA Consent checkbox, TCPA Consent Text: I consent to receive calls from [Business Name] at the telephone number provided above, including promotional calls. I understand that consent is not a condition of purchase.). The 'Save' and 'Cancel' buttons are at the bottom right.

Skill Transfer Module Properties

General Skills Announcements Priority Connectors **Queue Callback** Disposition

Initial Callback Number from: Queue Callback.Seed Number

Store Callback Number in: Queue Callback.Phone Number

☐ Allow collection and callback of International numbers?

Prompts

Playing the Phone Number: QC1YourPhoneNumberIs **Select**

Confirming the Phone Number: QC2PressOneIfCorrect **Select**

Entering the Phone Number: QC3PleaseEnterYourPhoneNumber **Select**

Recording Caller's Name: QC5PleaseClearlySayYourName **Select**

Confirmation and Thank You: QC4YouWillBeCalled **Select**

Timeouts

Disable Queue Callback after: Days: 1 Hours: 5 Minutes: 0 Seconds: 0

After Digit Timeout (secs): 5

TCPA Consent

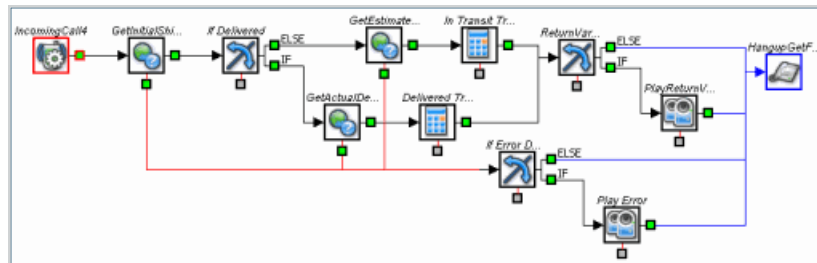
☐ TCPA Consent

TCPA Consent Text: I consent to receive calls from [Business Name] at the telephone number provided above, including promotional calls. I understand that consent is not a condition of purchase.

Save **Cancel**

Creating an IVR Script for Tracking a Shipment

This topic provides information on how to configure a campaign's IVR script with multiple Web Service Query modules. This example gets tracking and delivery information by sending a query with tracking identifiers to a Web service.



Creating Variables to Support Query Responses

The first step in creating this IVR script is to set the variables that you will be using as the interaction is processed. These script variables are required for the query responses.

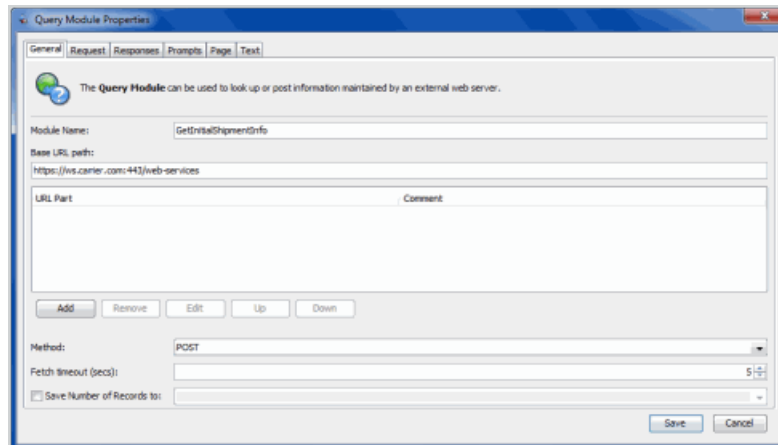
BrianW_FedExRouting Variables					
Variable	Type	Initial Value	Input	Output	Description
_BUFFER__	STRING		No	No	Default variable f...
_ExtContactFields__	KVLIST	0	Yes	Yes	Default variable f...
carrierActualDeliveryAddress	STRING		No	No	Actual delivery a...
debug	STRING		Yes	Yes	INPUT: Set to Y t...
carrierActualDeliveryAddressCity	STRING		No	No	Actual delivery a...
carrierActualDeliveryAddressState	STRING		No	No	Actual delivery a...
carrierActualDeliveryDate	STRING		No	No	Actual delivery d...
carrierActualDeliveryTime	STRING		No	No	Actual delivery t...
carrierEstDeliveryAddress	STRING		No	No	Address package...
carrierEstDeliveryAddressCity	STRING		No	No	City package wa...
carrierEstDeliveryAddressState	STRING		No	No	State package m...
carrierEstDeliveryDate	STRING		No	No	Estimated deliver...
carrierEstDeliveryTime	STRING		No	No	Estimated deliver...
carrierLocationAddressCity	STRING		No	No	Current location ...
carrierLocationAddressState	STRING		No	No	Current location ...
carrierResponse	STRING		No	No	XML response glob...
carrierShippedDate	STRING		No	No	Date package wa...
carrierShippedTime	STRING		No	No	Time package wa...
carrierTrackingAddress	STRING		Yes	Yes	RETURN: Estim...
carrierTrackingDeliveryDate	DATE		Yes	Yes	RETURN: Estim...
carrierTrackingDeliveryTime	TIME		Yes	Yes	RETURN: Estim...
carrierTrackingLocation	STRING		Yes	Yes	RETURN: Curren...
carrierTrackingNumber	STRING		Yes	Yes	INPUT: FedEx Tr...
carrierTrackingShippedDate	DATE		Yes	Yes	RETURN: Date p...
carrierTrackingShippedTime	TIME		Yes	Yes	RETURN: Time p...
carrierTrackingStatus	STRING		Yes	Yes	RETURN: "In Tra...

Adding and Configuring the Query Modules

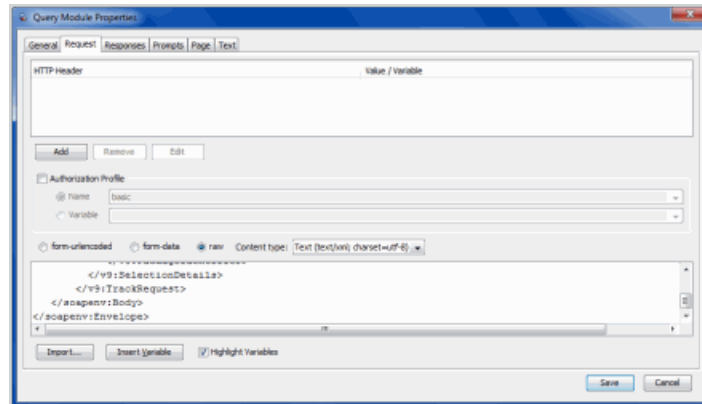
This IVR script uses three Query modules. Although each Query module uses the same SOAP query, each parses the response as needed for that stage of the IVR script.

Configuring the *GetInitialShipmentInfo* Query Module

The first Query module, named *GetInitialShipmentInfo*, gets the initial shipping information needed to proceed. This information includes the city and state from which the package was shipped and the actual date and time this shipment began.



- 1 Provide an intuitive name for this module.
- 2 Add the base URL path.
- 3 Assigns the POST method for this query.



- 4 Configure the Request tab to gather the initial shipment information by using the following code:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v9="http://carrier.com/ws/track/v9">
  <soapenv:Header/>
  <soapenv:Body>
    <v9:TrackRequest>
      <v9:WebAuthenticationDetail>
        <v9:UserCredential>
          <v9:Key>Mh6hPS34N5fCn4lB</v9:Key>

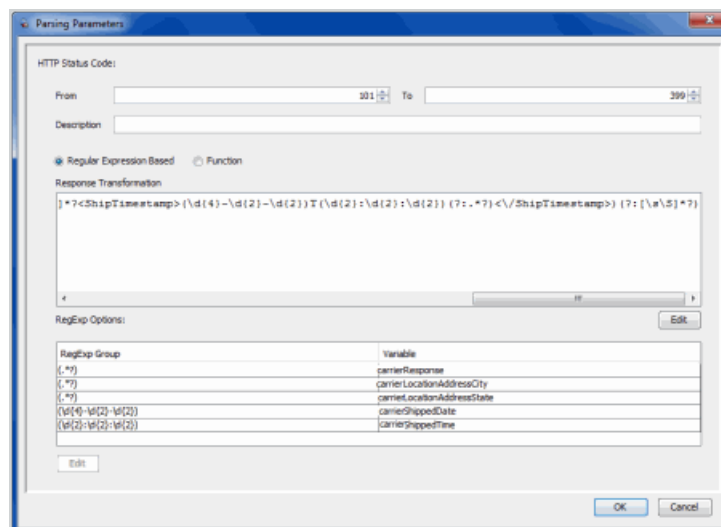
<v9:Password>bCp7cpvKpT7l9HuaVGi7f6jHn</v9:Password>
        </v9:UserCredential>
      </v9:WebAuthenticationDetail>
      <v9:ClientDetail>
        <v9:AccountNumber>123456789</v9:AccountNumber>
        <v9:MeterNumber>987654321</v9:MeterNumber>
        <v9:Localization>
          <v9:LanguageCode>EN</v9:LanguageCode>
          <v9:LocaleCode>US</v9:LocaleCode>
        </v9:Localization>
      </v9:ClientDetail>
      <v9:TransactionDetail>
        <v9:CustomerTransactionId>Track By Number_v9
          </v9:CustomerTransactionId>
        <v9:Localization>
          <v9:LanguageCode>EN</v9:LanguageCode>
          <v9:LocaleCode>US</v9:LocaleCode>
        </v9:Localization>
      </v9:TransactionDetail>
    </v9:TrackRequest>
  </v9:Body>
</v9:Envelope>
```

```

<v9:Version>
  <v9:ServiceId>trck</v9:ServiceId>
  <v9:Major>9</v9:Major>
  <v9:Intermediate>1</v9:Intermediate>
  <v9:Minor>0</v9:Minor>
</v9:Version>
<v9:SelectionDetails>
  <v9:PackageIdentifier>
    <v9:Type>TRACKING_NUMBER_OR_DOORTAG</v9:Type>
    <v9:Value>&&CarrierTrackingNumber&&</v9:Value>
  </v9:PackageIdentifier>
</v9:SelectionDetails>
</v9:TrackRequest>
</soapenv:Body>
</soapenv:Envelope>

```

The Responses tab parses the parameters into your predefined variables.



5 Parse the query response by using the following code:

```

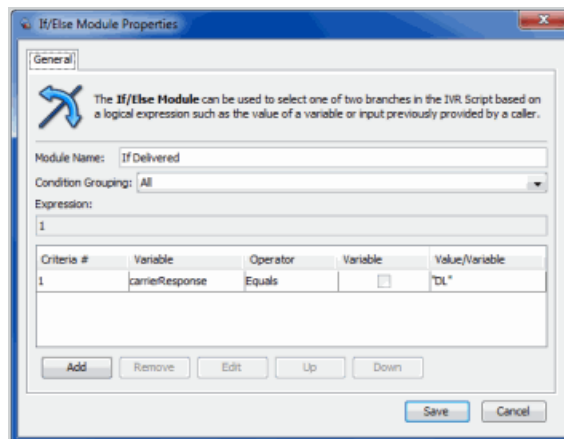
(?:<StatusDetail>[\\s\\S]*?<Code>(.*?)
</Code>[\\s\\S]*?<Location>[\\s\\S]*?<City>(.*?)
</City>[\\s\\S]*?<StateOrProvinceCode>(.*?)
</StateOrProvinceCode>[\\s\\S]*?
</StatusDetail>[\\s\\S]*?<ShipTimestamp>(\d{4}-\d{2}-\d{2})T(\d{2}:\d{2}:\d{2})(?:.*?</ShipTimestamp>)?(?:[\\s\\S]*?)
</ShipTimestamp>)(?:[\\s\\S]*?)

```

The remaining tabs in this first query module may remain unused.

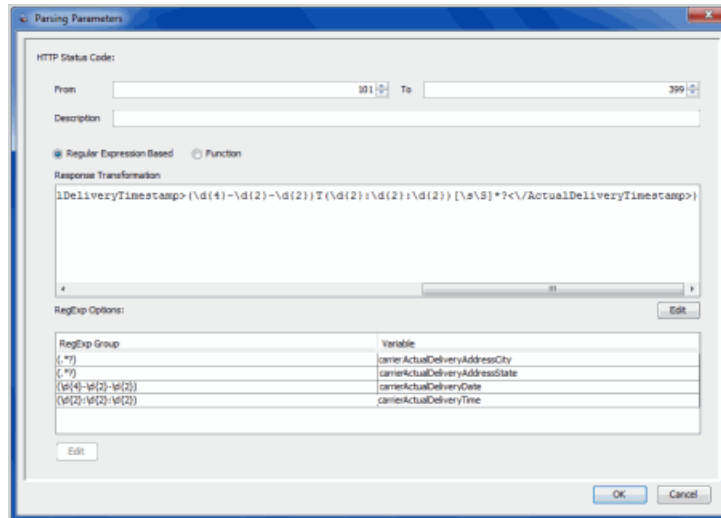
Adding and Configuring the *IfDelivered* If/Else Module

This second module in this IVR script uses the returned variables from the first **Query** module to determine if the package has been delivered. Using the **If/Else** module enables separate distribution branches that lead to specific **Query** modules. If the item has been delivered, the If branch is taken and the call is delivered to the follow-on **Query** module and subsequent **Set Variable** module to retrieve and assign the delivery information. If the item is still in transit, the Else branch delivers this call to an alternate follow-on **Query** module and subsequent **Set Variable** module to retrieve and assign the estimated delivery information.



Adding and Configuring the *ActualDeliveryInfo* Branch

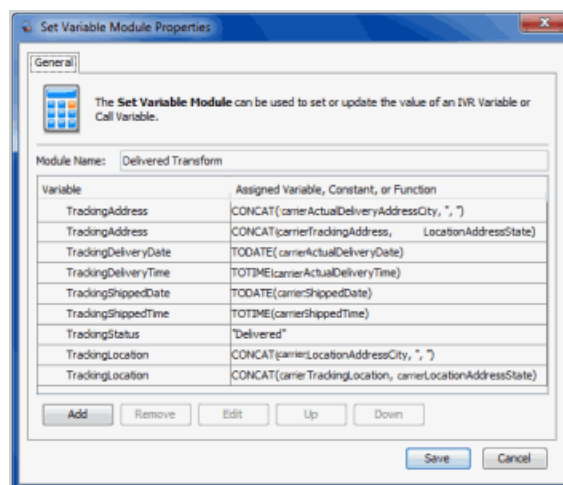
When the item has been delivered, the actual delivery branch retrieves and assigns the delivery information.



- 1 Configure this module similarly to the first Query module with this code:

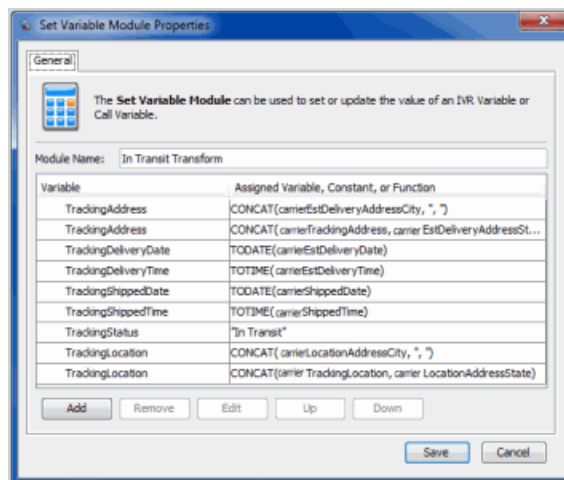
```
(?:<ActualDeliveryAddress>[\\s\\S]*?<City>(.*?)</City>
[\\s\\S]*?<StateOrProvinceCode>(.*?)</StateOrProvinceCode>
[\\s\\S]*?<ActualDeliveryTimestamp>(\d{4}-\d{2}-\d{2})T(\d{2}:\d{2}:\d{2})[\\s\\S]*?</ActualDeliveryTimestamp>)
```

- 2 Configure the **SetVariable** module, *DeliveredTransform*, to assign the variables retrieved in the previous **Query** Module.



Adding and Configuring the *GetEstimatedDeliveryInfo* Branch

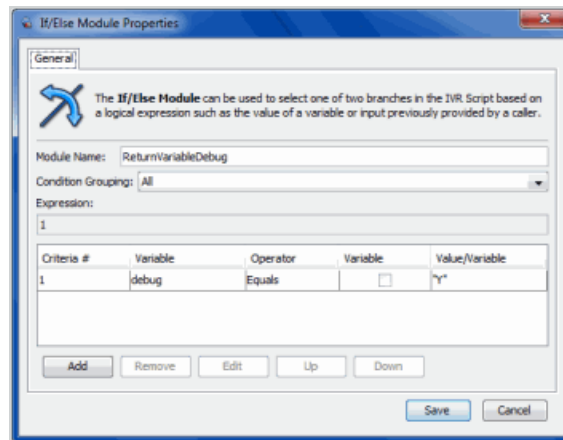
This branch uses a **Query** module and **Set Variable** module to retrieve the estimated delivery information. The setup is similar to the *IfDelivered* branch above, with the exception that the TrackingStatus is set to InTransit, not Delivered.



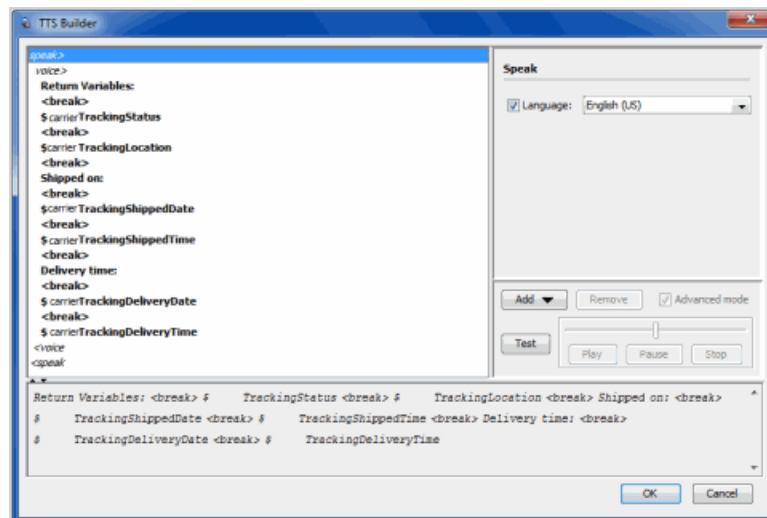
Adding and Configuring the *ReturnVariableDebug* Module

You configure this branch using an **If/Else** module to check for the presence of a return variable and a **Play** module to play any error message to the caller.

- 1 Add an If/Else module, you check the value of your debug variable.



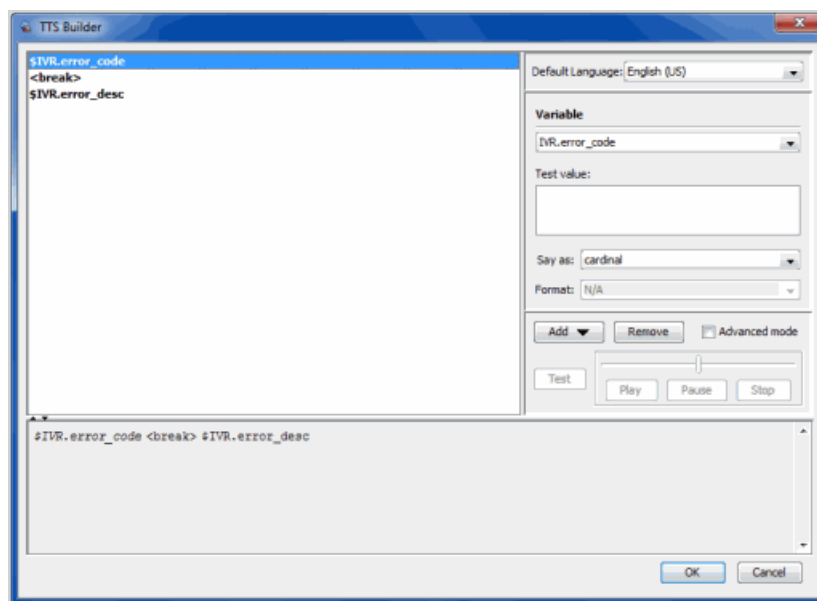
- 2 Add criteria to check for the Debug variable is set to "Y".
- 3 Configure the IF branch to play that variable recording to the caller.
- 4 Add a Play module to play TTS as described below.



- 5 Configure the Else branch to proceed to disconnect if there are no error messages to play.

Adding and Configuring the *IfErrorDebug* Module

You configure the final **If/Else** module to capture any IVR errors from any of the prior **Query** modules. The entry into this module is the error branch from any of the three **Query** modules that you have configured. You configure this branch similarly to the error branch above, except the TTS configuration for the final **Play** module, shown below.



This completes your configuration for this IVR script.

Using an Agent Extension in an IVR

This section shows how to create a script that allows callers to make a menu selection or enter an extension without going to a second prompt with the Get Digits and the Play module. The caller can press 1 for Sales, 2 for Support, or enter an extension. All extensions start with the number 9.

Your IVR script should be created and the IVR Flow Editor window be open.

Configuring the Get Digits Module

We use the Get Digits module to accept the values entered by the caller. You can also add the Extension Transfer Module instead of the Get Digits. The Extension Transfer Module will accept the digits entered by the caller, and transfer the call. The Extension Transfer Module can play the default prompts. Read more in Extension Transfer Module.

- 1 Add a Get Digits module to the workspace.
- 2 Open the **Module Properties**.
- 3 Set these properties.
 - a Select the default BUFFER variable.
 - b Check **Clear User Input**.

The screenshot shows the 'Get Digits Module Properties' dialog box. The 'General' tab is active. The 'Module Name' is 'GetDigits4'. The 'Target Variable' is '_BUFFER_'. The 'Input Format' is empty. The 'Maximum Number of Digits' is 4. The 'Termination Digit' is '#'. The 'Max Silence (secs)' is 2. The 'Max Time to Enter Digits (secs)' is 20. The 'Clear User Input' checkbox is checked and highlighted with a red box. The 'Collapsible' checkbox is unchecked. There are 'Save' and 'Cancel' buttons at the bottom right.

- 4 Click **Save**.

Configuring the Case Module

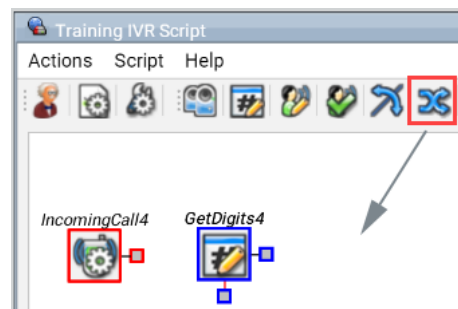
The Case module is used to split the script flow based on the user input to the branches. This module has a No Match branch which is used when the user input does not match any criteria defined in the Case module.

This script has four branches: Sales, Support, Extension, and No Match:

- The caller can press 1 for sales.
- The caller can press 2 for support or enter an extension.

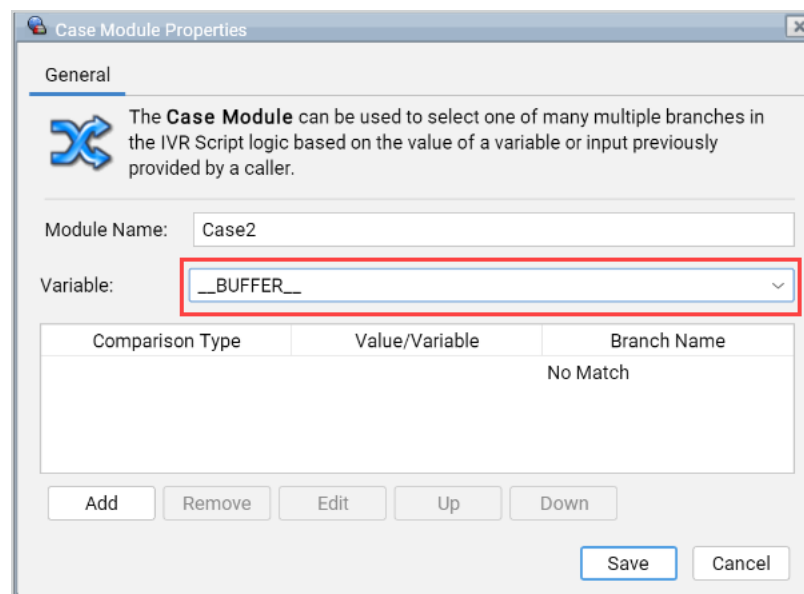
- The caller can enter an extension. All extensions start with 9. You need to enter a wild card % to substitute for any other character or characters in a string.
- If the caller does not enter a number that matches Sales, Support, or Extension, the module selects No Match. Link the No Match according to your needs.

1 Add a Case module to the workspace.



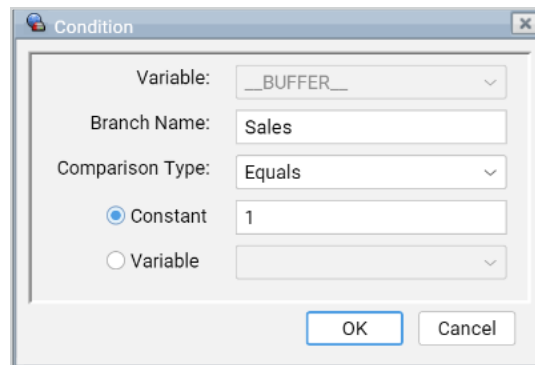
2 Open the **Module Properties**.

- 3** In the Variable field, select the variable that contains the caller's input, in this case: BUFFER.



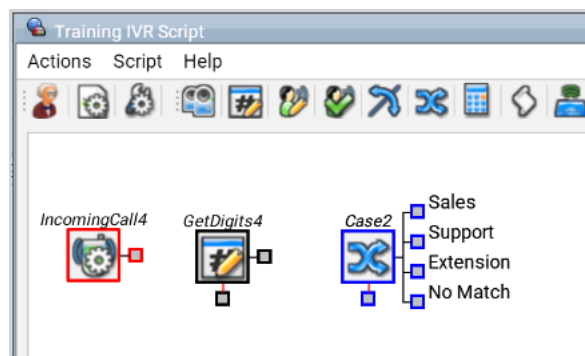
- 4** To add the first branch, Sales, click **Add**.

- a Enter Sales into the Branch Name text field, select Equals in the Comparison Type, and enter 1 as the constant.



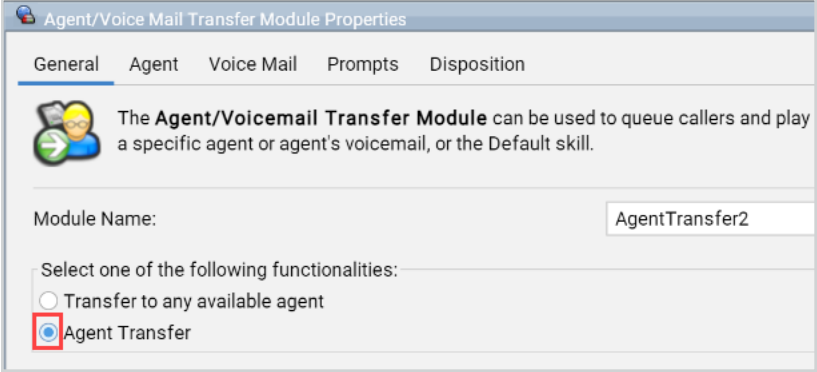
- b Click **OK**.

- 5 Repeat step 4 to create a branch named Support, with 2 as the constant.
- 6 Repeat step 4 to create a branch named Extension, with the Like operator.
You need to enter a wild card “%” used to substitute for any other character or characters in a string.
- 7 Click **Save**.



Configuring the Agent/Voice Mail Transfer Module

- 1 Add a Agent/Voice Mail Transfer module to the workspace.
- 2 Open the **Module Properties**.
- 3 In the General tab, select **Agent Transfer**.



The **Agent/Voice Mail Transfer Module** can be used to queue callers and play a specific agent or agent's voicemail, or the Default skill.

Module Name:

Select one of the following functionalities:

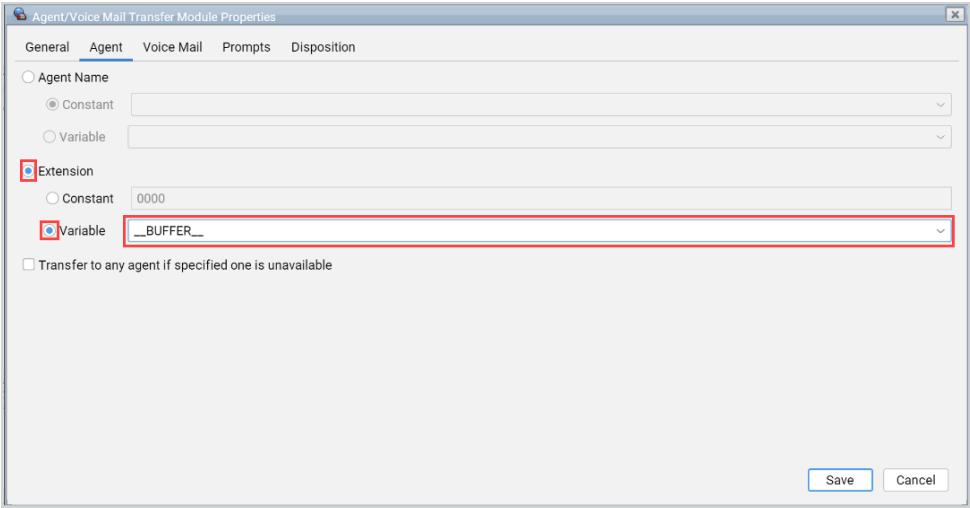
☐ Transfer to any available agent

☒ Agent Transfer

4 Select the Agent tab.

5 Select the default BUFFER option.

The Agent/Voicemail Transfer Module uses the BUFFER variable to find the Agent's extension.



General Agent Voice Mail Prompts Disposition

☐ Agent Name

☒ Extension

☐ Constant

☒ Variable

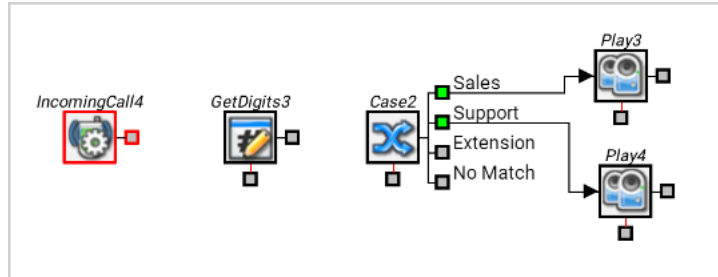
☐ Transfer to any agent if specified one is unavailable

Save Cancel

6 Click **Save**.

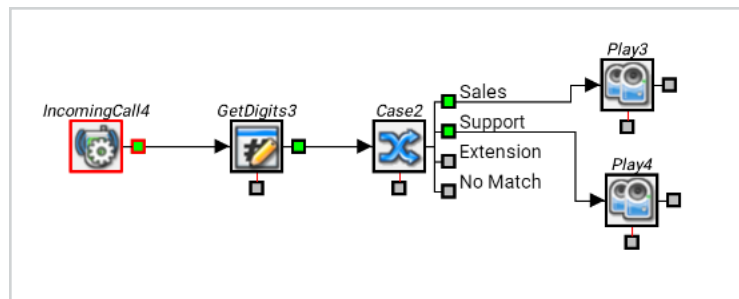
Adding Modules for Each Case

Depending on your campaign requirements, add the modules for each Case Module branch. In our example, we added the Play modules for the Sales and Support branches. For the Extension branch, we added the Agent Transfer module, and the Play module.

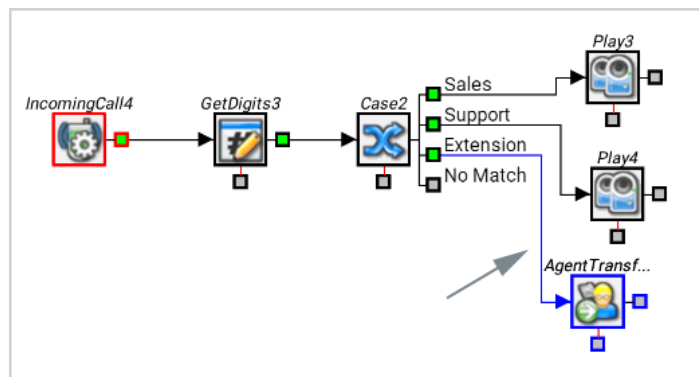


Connecting the Modules

- 1 Connect the modules in this sequence: **Incoming > Get Digits > Case**.

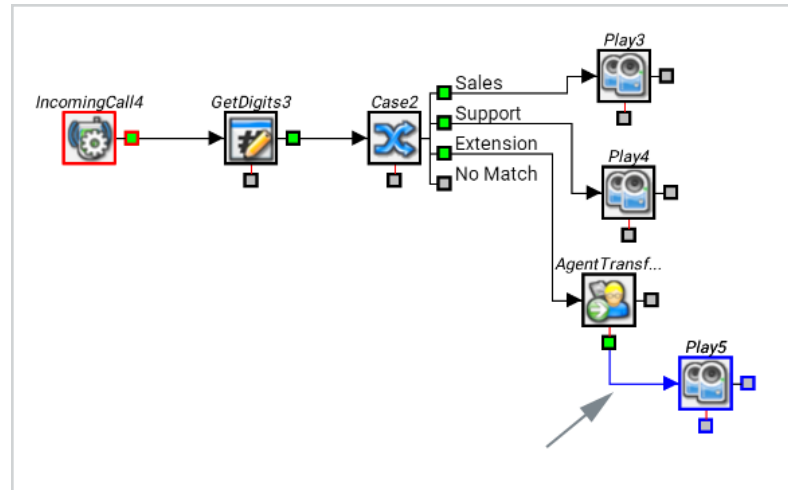


- 2 Connect the Extension branch to the Agent/Voice Mail Transfer module.

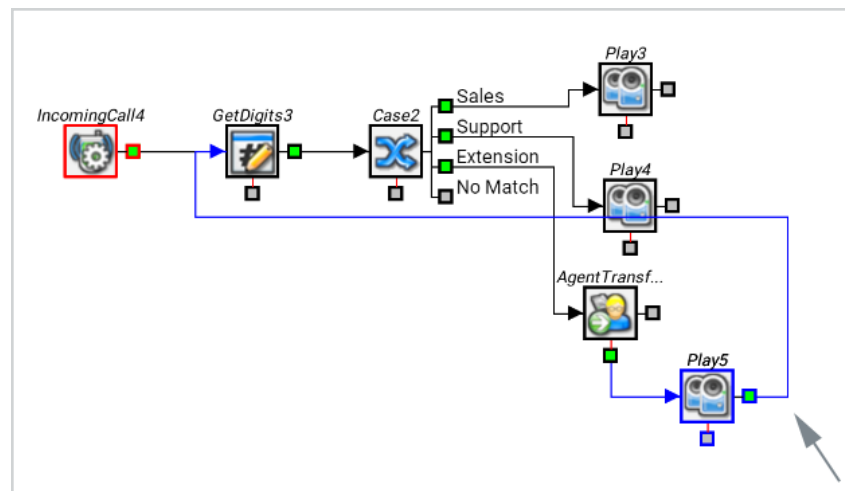


- 3 Add a new Play module and connect it to the extension port of Agent/Voice Mail Transfer module.

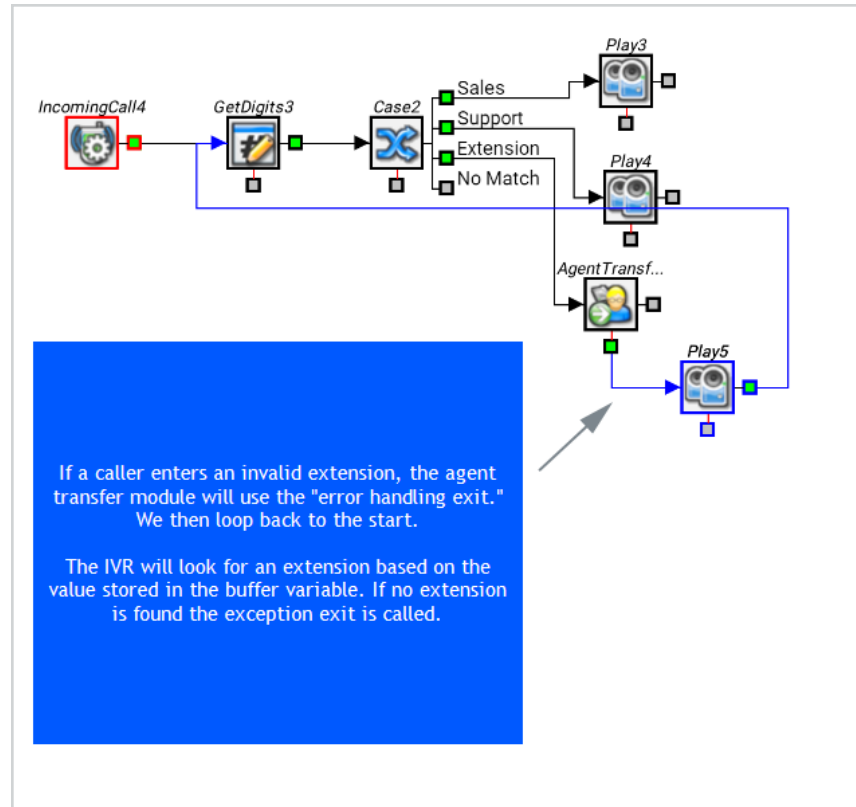
If a caller enters an invalid extension, the Agent/Voice Mail Transfer module uses the Error handling exit.



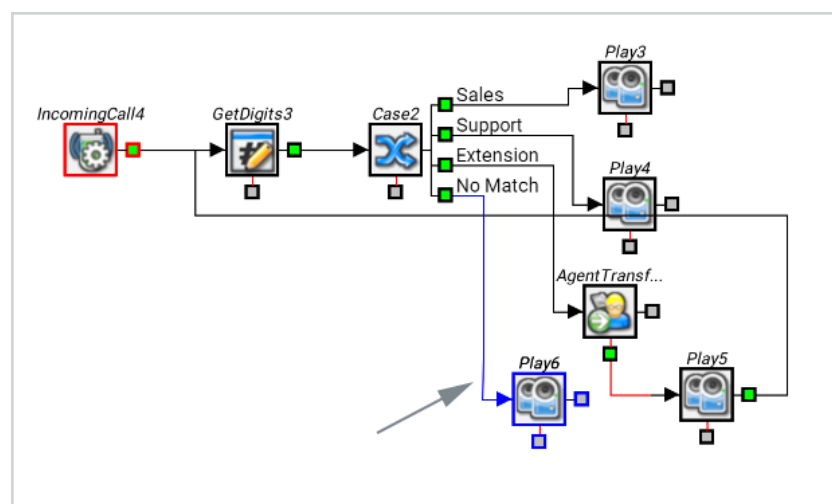
- 4 Loop back the third Play module to the start (the Get Digits module).



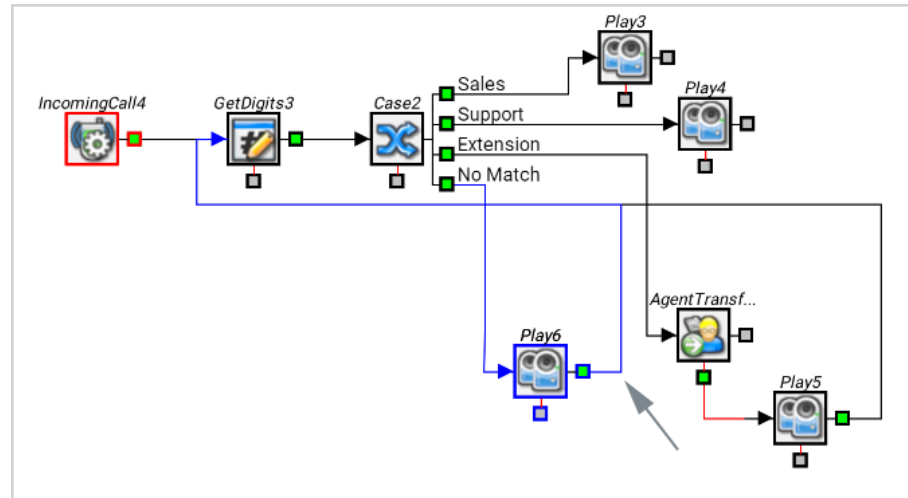
The following image shows what happens when a caller enters an invalid extension.



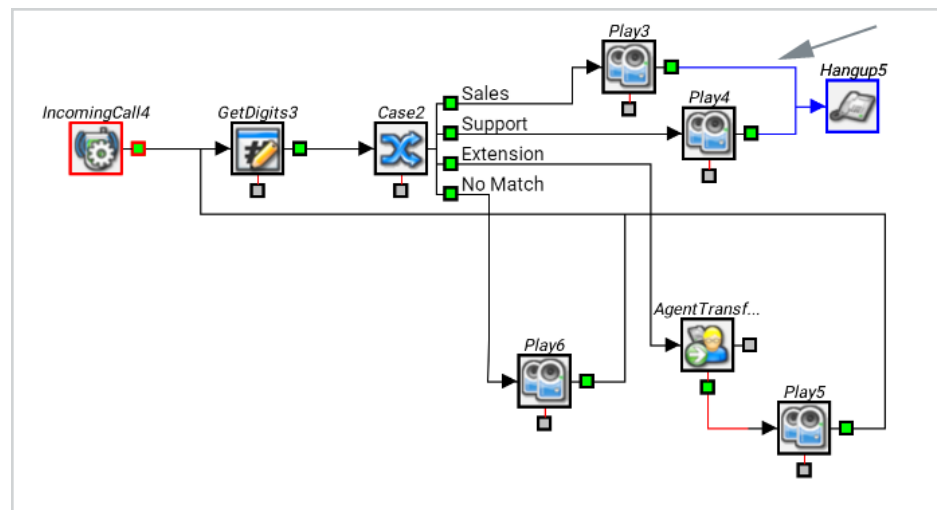
- 5 Add one more Play module, and connect the No Match branch of the Case module to the new Play module.



- 6 Loop back the fourth Play module to the beginning (the Get Digits module).



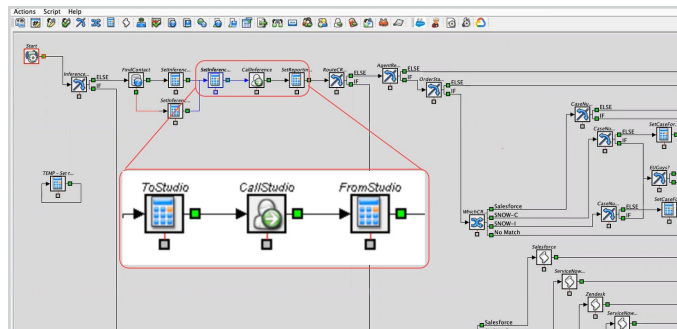
- 7 Connect the first and the second Play modules to the Hang Up module.



Incoming Call Transfer to Studio IVA

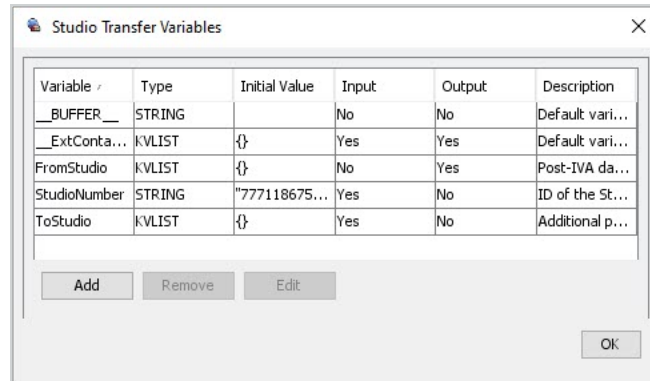
This example identifies some basic interaction and data transfer using the Five9 Studio IVA solution. Your particular IVR script will involve more modules than discussed here. The focus of this example is three specific modules:

- **Set Variable** module to collect data to pass to Studio.
- **Third-party Transfer** module to transfer the interaction and collected data to Studio.
- **Set Variable** module to pass return data to VCC, such as customer journey details and interaction information.



Preparing Your Script Variables

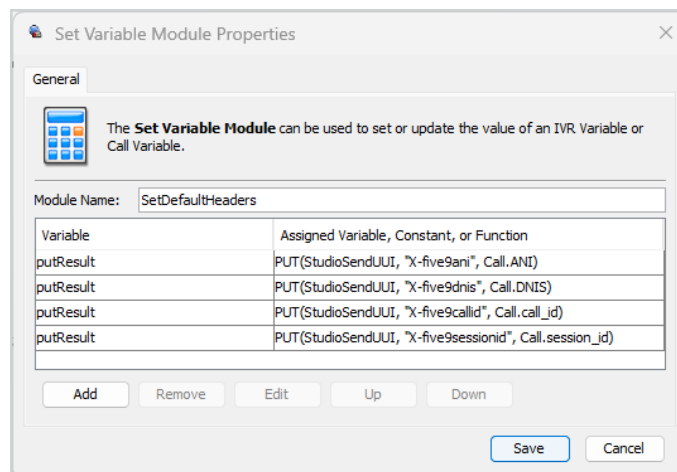
You define and modify custom variables using [Script Variables](#). The following variables support the modules identified in this example.



- 1 FromStudio:** This variable stores the return data following the IVA interaction. This data is defined in the Return From IVA, or second Set Variable module used in this example.
- 2 StudioNumber:** This variable translates the 12-digit Studio number for use in Studio agent contact.
- 3 ToStudio:** Data gathered in your IVR script to send to the IVA with the interaction.

Collecting Data for IVA (to Studio)

The interaction heading toward Five9 IVA comes from the [Five9 Engagement Workflow](#) embedded in your campaign IVR script. Once the interaction is introduced to this campaign, the first [Set Variable Module](#) captures and stores information from Five9 to send with the interaction, such as DNIS. This module example explains capturing that DNIS information and lists additional, commonly used variables.



- 1 Drag a Set Variable module into your IVR script canvas and double-click the module to edit the properties.
- 2 The Assignment window opens. Click **Add** to create a new variable.

Assignment

Variable: _BUFFER_

Type: STRING

Description: Default variable for collecting digits

Assigned Value

☐ Constant ☐ Variable ☒ Function ☐ NULL

PUT(KVLIST, STRING, STRING)

Function Arguments:

#1	Variable	ToStudio
#2	Constant	X-five9dnis
#3	Variable	Call.DNIS

OK Cancel

- 3 In the Variable dropdown menu, select _BUFFER_. You can select from the script variables created in [Preparing Your Script Variables](#).
- 4 Select the Function radio button for Assigned Value and select PUT (KVLIST, STRING, STRING) from the Function dropdown. Function arguments are as follows:

a Argument #1: Variable - ToStudio

This calls the ToStudio script variable to store the Five9 DNIS and send it to Studio.

b Argument #2: Constant - X-five9dnis <insert your variable name here>

Important: Variable names exactly match those in Studio. This example is not using a required variable name. Use all lowercase characters for variable names because the data that returns in the IVA interaction is always lowercase and the names must match

c Argument #3: Variable - Call.DNIS

This value is the Five9 call DNIS and is selectable in the **Call Variables** section of the variable dropdown.

The screenshot shows the 'Assignment' dialog box with the following details:

- Variable:** `_BUFFER_`
- Type:** `STRING`
- Description:** Default variable for collecting digits
- Assigned Value:** `PUT(KVLIST, STRING, STRING)`
- Function Arguments:**

#	Type	Value
#1	Variable	ToStudio
#2	Constant	X-five9dnis
#3	Variable	Call.DNIS

A dropdown menu is open for the third argument, showing a list of **Call Variables** including `Call.ANI`, `Call.DNIS` (which is highlighted), `Call.bill_time`, `Call.call_id`, `Call.campaign_id`, and `Call.campaign_name`.

- 5 Click **OK** to save the variable definition.

Use this process to create other commonly used variables shown in this script, such as `Call.session_id` and `Call.call_id`.

Sending the Interaction to Studio IVA

The second module in this sequence, a [Third Party Transfer Module](#), sends this interaction to the Studio IVA. This module uses the **StudioNumber**, **FromStudio**, and **ToStudio** variables.

The 3rd Party Transfer Module Properties dialog box is shown. It has tabs for General, Prompts, and Disposition. The General tab is active. The description states: "The 3rd-Party Transfer Module can be used to blind-transfer (single-step transfer) a caller to a specified phone number." The Module Name is set to "CallStudio". The Third Party Number is set to "Variable" with a dropdown menu showing "StudioNumber". The Recording section has "Record 3rd Party call" unchecked and "Return After 3rd Party Call" checked. Under "Return After 3rd Party Call", "Send Data to 3rd Party" is checked with a dropdown set to "ToStudio", "Receive Data from 3rd Party" is checked with a dropdown set to "FromStudio", "Max Time of 3rd Party Call (Seconds)" is set to 300, "Digit to Interrupt 3rd Party Call & Return" is set to a dropdown, and "Ringing Timeout (Seconds)" is set to 10. There are Save and Cancel buttons at the bottom right.

Important: You must select **Return After 3rd Party** to retrieve the interaction and select **Send Data to 3rd Party** and **Return Data From 3rd Party** to exchange variable information.

Data Return (From Studio)

The interaction remains in Studio until it is resolved or passed to a live agent. You can define interaction data to return in your [Studio Header Node](#). Interaction information is captured from SIP headers as [XSIP variables](#). The second Set Variable module identifies the return information from Studio. This example shows retrieving the customer journey information.

The Set Variable Module Properties dialog box is shown. It has a General tab. The description states: "The Set Variable Module can be used to set or update the value of an IVR Variable or Call Variable." The Module Name is set to "FromStudio". Below is a table with two columns: "Variable" and "Assigned Variable, Constant, or Function". The first row is highlighted in blue and contains "_BUFFER_" and "GET(FromStudio, 'X-customer-journey')". Below the table are buttons for Add, Remove, Edit, Up, and Down. There are Save and Cancel buttons at the bottom right.

Variable	Assigned Variable, Constant, or Function
BUFFER	GET(FromStudio, "X-customer-journey")

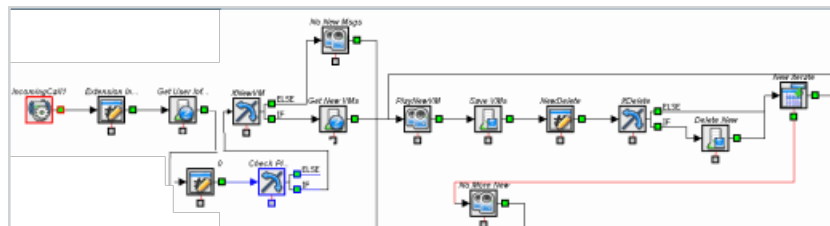
Managing Voicemail

This example is divided into specific functional branches for construction. Each branch loops back to a main menu module to enable multiple options in a single IVR script. This script includes the following branches.

[Retrieving Voicemail Messages](#)
[Configuring Voicemail Account for New Users](#)
[Playing the Current Voicemail Greeting](#)
[Recording a Voicemail Greeting](#)
[Changing an Existing Voicemail PIN](#)

Retrieving Voicemail Messages

Configure this branch to enable voicemail account access. Use the following modules to identify the caller's extension and credentials. The modules are connected in series, in the order listed.



Get Digits Module

Use this module to collect caller input. This input identifies the user using their 4-digit Five9 extension.

The screenshot shows the 'Get Digits Module Properties' dialog box with the 'Prompts' tab selected. The 'General' tab is also visible. The 'Exit The Module in Case of A Prompt Exception' checkbox is unchecked. The 'Type' column contains 'TTS' and the 'Text / File Name' column contains 'What is your extension?'. The 'Add', 'Remove', 'Edit', 'Up', 'Down', and 'Convert to ML Prompt' buttons are visible at the bottom. The 'Save' and 'Cancel' buttons are at the bottom right.

Set the maximum number of digits in the BUFFER variable to 4.

The screenshot shows the 'Get Digits Module Properties' dialog box with the 'General' tab selected. The 'Prompts' tab is also visible. The 'Module Name' is 'Extension Input 0'. The 'Target Variable' is 'InExt'. The 'Input Format' is set to 'DTMF'. The 'Maximum Number of Digits' is set to 4. The 'Termination Digit' is '#'. The 'Max Silence (secs)' is 10. The 'Max Time to Enter Digits (secs)' is 20. The 'Clear User Input' checkbox is checked. The 'Collapsible' checkbox is unchecked. The 'Save' and 'Cancel' buttons are at the bottom right.

System Info Module

This module retrieves the parameters associated with the extension identified in the previous step. Configure this module to populate the script variables and establish the specific user information associated with the extension entered in the previous module.

The **System Info Module** can be used to look up Users and Skills based on their properties such as name or extension, to select a destination for the call.

Module Name:

System Object:

Lookup Condition:

Returned Parameters:

Parameter	Variable
First Name	FirstName
Last Name	LastName
PIN	UserPIN
Greeting	Greeting

If Multiple Records Match the Condition:

☐ Raise an Exception

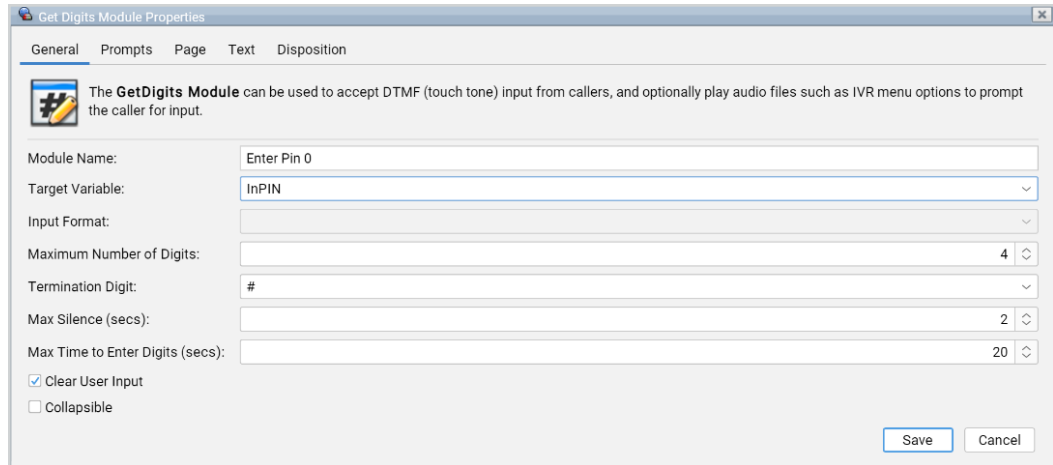
☐ Save Number of Records to:

Get Digits Module

Use this module to collect the user input and authenticate the PIN associated with the user and extension. The maximum number of digits in the BUFFER variable is 4.

☐ Exit The Module in Case of A Prompt Exception

Type	Text / File Name
TTS	Enter your four digit pin.



The **Get Digits Module** can be used to accept DTMF (touch tone) input from callers, and optionally play audio files such as IVR menu options to prompt the caller for input.

Module Name: Enter Pin 0

Target Variable: InPIN

Input Format:

Maximum Number of Digits: 4

Termination Digit: #

Max Silence (secs): 2

Max Time to Enter Digits (secs): 20

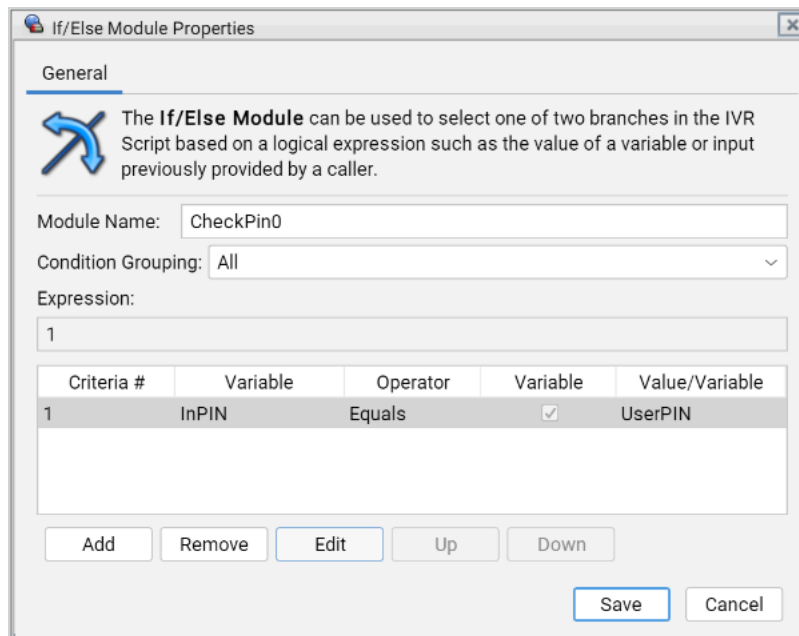
☒ Clear User Input

☐ Collapsible

Save Cancel

If/Else Module

Use this module to verify the PIN entered against the PIN retrieved from system information. If the entered PIN matches the `InPIN` variable, the call proceeds to voicemail retrieval. If the PIN entered does not match the call returns to the enter PIN step. If the `InPIN` variable is null, indicating that a PIN has not yet been created for this extension, the caller should be directed to the [Configuring Voicemail Account for New Users](#) branch of this script. For more information, see [Configuring Voicemail Account for New Users](#).



The **If/Else Module** can be used to select one of two branches in the IVR Script based on a logical expression such as the value of a variable or input previously provided by a caller.

Module Name: CheckPin0

Condition Grouping: All

Expression:

1

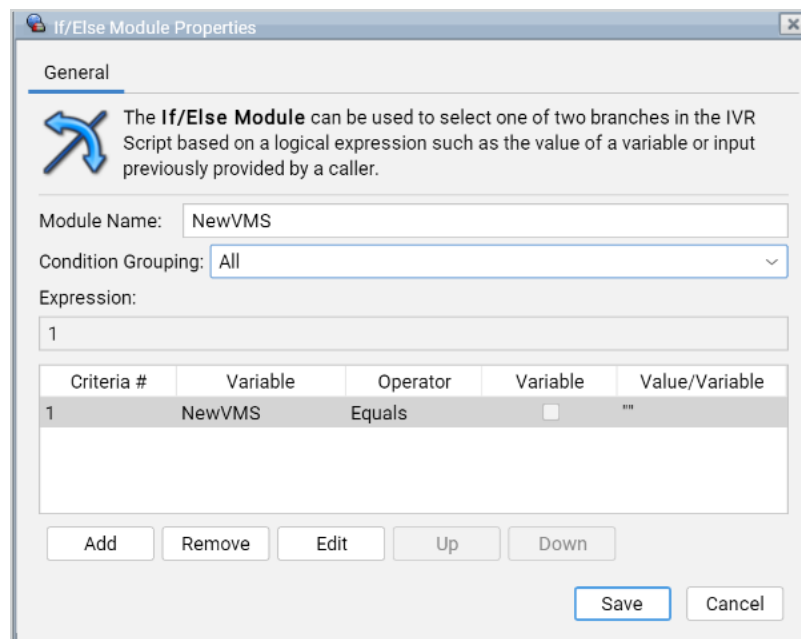
Criteria #	Variable	Operator	Variable	Value/Variable
1	InPIN	Equals	☒	UserPIN

Add Remove Edit Up Down

Save Cancel

If/Else Module

Use this module to check the `NewVMS` variable to check for new voicemail messages. This variable has an initial value of 0 and any value greater than zero indicates new voicemail messages. If new messages exist the call proceeds to retrieve these new messages.



The screenshot shows the 'If/Else Module Properties' dialog box with the 'General' tab selected. It contains a description of the module's function, input fields for 'Module Name' and 'Condition Grouping', an 'Expression' field, and a table for defining criteria. The 'Module Name' is set to 'NewVMS' and 'Condition Grouping' is set to 'All'. The 'Expression' field contains the number '1'. The table has one row with 'Criteria # 1', 'Variable NewVMS', 'Operator Equals', 'Variable' (empty), and 'Value/Variable' (empty). Buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', 'Save', and 'Cancel' are at the bottom.

General

The **If/Else Module** can be used to select one of two branches in the IVR Script based on a logical expression such as the value of a variable or input previously provided by a caller.

Module Name:

Condition Grouping:

Expression:

Criteria #	Variable	Operator	Variable	Value/Variable
1	NewVMS	Equals		

System Info Module

Use this module to retrieve new voicemail messages to play in the next step.

The **System Info Module** can be used to look up Users and Skills based on their properties such as name or extension, to select a destination for the call.

Module Name:

System Object:

Lookup Condition:
(Extension = InPIN) AND (Is New = 1)

Returned Parameters:

Parameter	Variable
Number	phone
Creation Date	date
Creation Time	time
File name	filename

If Multiple Records Match the Condition:

☐ Raise an Exception

☐ Save Number of Records to:

Play Module

Use this module to play the date and time of each voicemail message, and the associated voicemail message file, identified in the previous step. Additional messages are played in the order they were received.

Play Module Properties

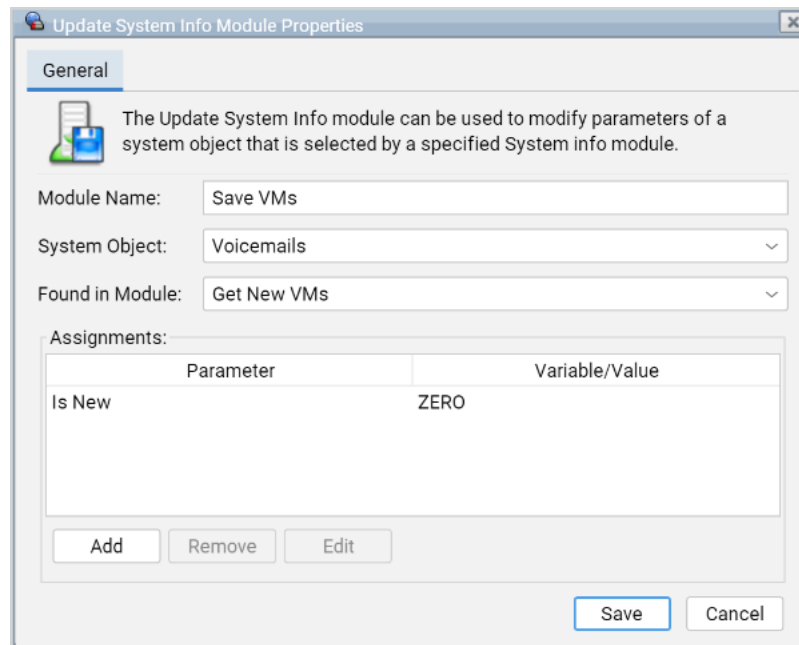
General **Prompts** Page Text Disposition Email Reply

Type	Text / File Name
TTS	Message and Pause(medium) Created \$date at \$time Pause(medium)
File's Variable	filename

Update System Info Module

Use this module to reset the `IsNew` variable to 0 for the voicemail, marking that message as played and removing that message from the list of messages flagged as

new voicemail messages.



The Update System Info module can be used to modify parameters of a system object that is selected by a specified System info module.

Module Name: Save VMs

System Object: Voicemails

Found in Module: Get New VMs

Assignments:

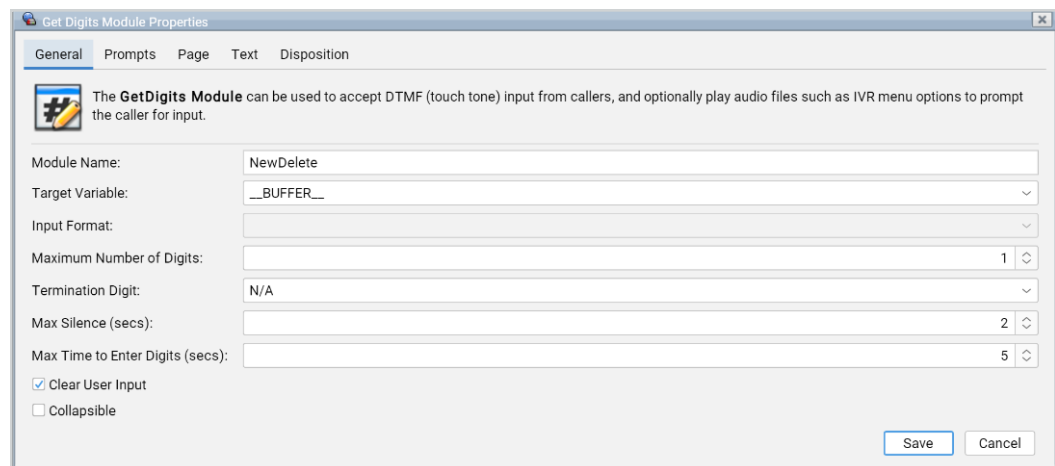
Parameter	Variable/Value
Is New	ZERO

Add Remove Edit

Save Cancel

Get Digits Module

Use this module to present delete or save options to the caller for the message that was just played. Configure this module to accept a maximum of one digit.



The **GetDigits Module** can be used to accept DTMF (touch tone) input from callers, and optionally play audio files such as IVR menu options to prompt the caller for input.

Module Name: NewDelete

Target Variable: __BUFFER__

Input Format:

Maximum Number of Digits: 1

Termination Digit: N/A

Max Silence (secs): 2

Max Time to Enter Digits (secs): 5

☒ Clear User Input

☐ Collapsible

Save Cancel

Configure TTS to prompt the caller to press seven to delete the message or nine to save the message.

The screenshot shows the 'Play Module Properties' dialog box with the 'Prompts' tab selected. The dialog has tabs for General, Prompts, Page, Text, Disposition, and Email Reply. The main area contains a table with two columns: 'Type' and 'Text / File Name'. There is one row with 'TTS' in the 'Type' column and 'To delete this message, press seven. To save, press nine.' in the 'Text / File Name' column. Below the table are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', 'Convert to ML Prompt', 'Save', and 'Cancel'.

Type	Text / File Name
TTS	To delete this message, press seven. To save, press nine.

If/Else Module

Use this module to take action on the caller's save or delete choice. Configure the **If** branch to flow into the next **Update System Info** module when the caller chooses seven to delete the voicemail message. Configure the **Else** branch to go directly into the Iterator module to retrieve the next voicemail message.

The screenshot shows the 'If/Else Module Properties' dialog box with the 'General' tab selected. The dialog contains a description of the module, a 'Module Name' field set to 'IfDelete', a 'Condition Grouping' dropdown set to 'All', and an 'Expression' field set to '1'. Below these is a table with columns: Criteria #, Variable, Operator, Variable, and Value/Variable. There is one row with '1' in the 'Criteria #' column, '__BUFFER__' in the 'Variable' column, 'Equals' in the 'Operator' column, an empty checkbox in the 'Variable' column, and '7' in the 'Value/Variable' column. At the bottom are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', 'Save', and 'Cancel'.

The **If/Else Module** can be used to select one of two branches in the IVR Script based on a logical expression such as the value of a variable or input previously provided by a caller.

Module Name: IfDelete

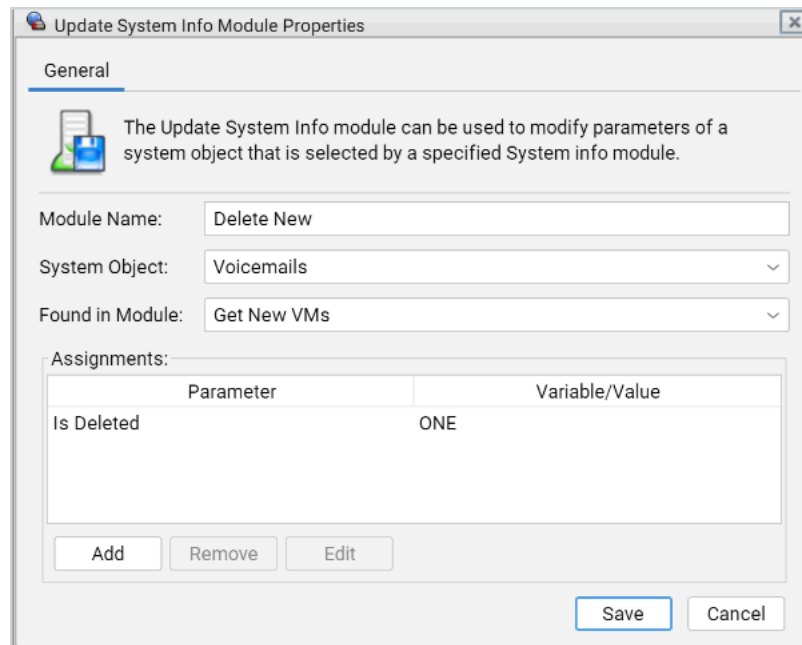
Condition Grouping: All

Expression: 1

Criteria #	Variable	Operator	Variable	Value/Variable
1	__BUFFER__	Equals	☐	7

Update System Info Module

Use this module to delete a voicemail message when the caller selects that option. Configure this module to feed into the **New Iterator** module to retrieve the next message. When the caller selects nine, to save the voicemail message, this module is bypassed and the previous **Get Digits** module flows directly into the next **New Iterator** module.



The screenshot shows the 'Update System Info Module Properties' dialog box with the 'General' tab selected. The dialog contains a description of the module's function, three input fields for configuration, an assignments table, and buttons for 'Add', 'Remove', 'Edit', 'Save', and 'Cancel'.

General

The Update System Info module can be used to modify parameters of a system object that is selected by a specified System info module.

Module Name:

System Object:

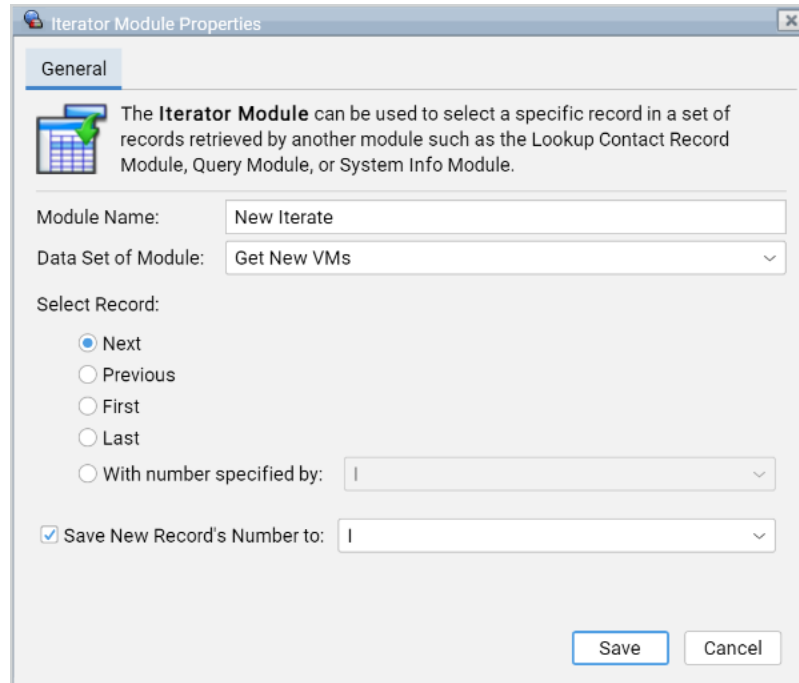
Found in Module:

Assignments:

Parameter	Variable/Value
Is Deleted	ONE

New Iterator Module

Use this module to advance to the next voicemail message. The output of this module loops back to the input side of the **Play** module configured to play new voicemail messages. Configure the exception branch of this module to a play module that informs the caller there are no more new voicemail messages.



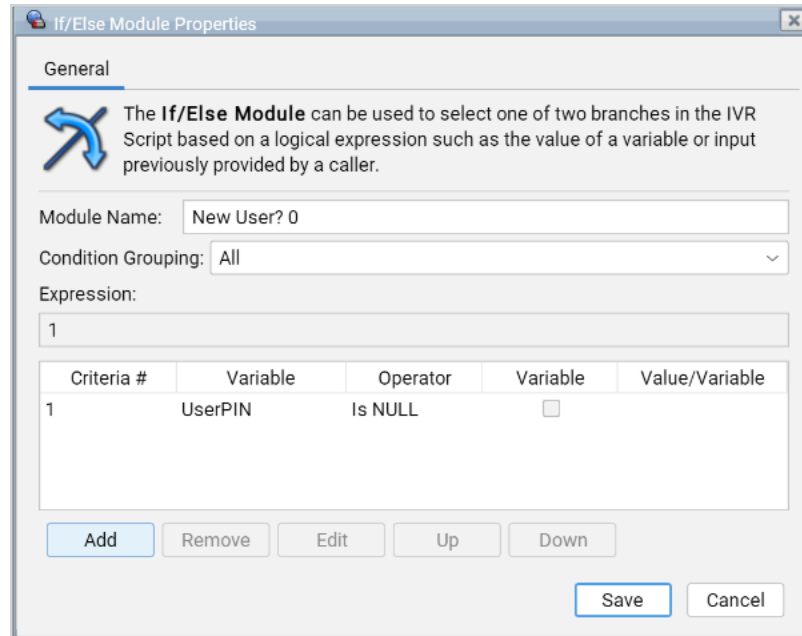
This is the end of this branch. Connect the output of this branch to the input of the You can choose to incorporate this branch into a more comprehensive IVR script and return the caller to the options menu, or send the output of your final play module to a hangup module to complete this script.

Configuring Voicemail Account for New Users

Configure this branch for new user setup. The input to this branch is the output of the first **System Info** module (see [System Info module](#)), which establishes the specific user information. The If/Else module that begins this branch evaluates the `UserPIN` parameter to identify new users. A null value for this parameter indicates an extension for which setup has not yet been completed.

If/Else Module

Use this module to determine whether the caller is a new user that needs to set up their voicemail PIN and greeting. This is accomplished by evaluating the `INPIN` variable. A null value indicates that the mailbox has not yet been configured.

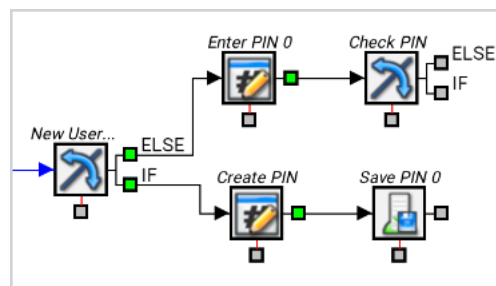


The screenshot shows the 'If/Else Module Properties' dialog box. The 'General' tab is selected. It contains a description of the module, a 'Module Name' field set to 'New User? 0', a 'Condition Grouping' dropdown set to 'All', and an 'Expression' field set to '1'. Below these is a table with one row of criteria.

Criteria #	Variable	Operator	Variable	Value/Variable
1	UserPIN	Is NULL	☐	

At the bottom of the dialog are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', 'Save', and 'Cancel'.

The **If** branch flows into the next Get Digits module to enable the caller to set their PIN. The **Else** branch goes to an alternate Get Digits module to gather the user PIN for verification into their voicemail account.



Get Digits Module

Use this module to prompt the caller to enter a 4-digit PIN.

The screenshot shows the 'Get Digits Module Properties' dialog box with the 'Prompts' tab selected. The 'General' tab is also visible. The 'Exit The Module in Case of A Prompt Exception' checkbox is unchecked. The 'Type' column contains 'TTS' and the 'Text / File Name' column contains 'Enter four digits you'd like set as your pin.' Below the table are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', 'Convert to ML Prompt', 'Save', and 'Cancel'.

Type	Text / File Name
TTS	Enter four digits you'd like set as your pin.

Store this new PIN in the `InPIN` variable.

The screenshot shows the 'Get Digits Module Properties' dialog box with the 'General' tab selected. The 'Prompts' tab is also visible. The 'Module Name' is 'Create PIN', 'Target Variable' is 'InPIN', 'Input Format' is empty, 'Maximum Number of Digits' is 4, 'Termination Digit' is '#', 'Max Silence (secs)' is 2, and 'Max Time to Enter Digits (secs)' is 20. The 'Clear User Input' checkbox is checked, and the 'Collapsible' checkbox is unchecked. Below the fields are 'Save' and 'Cancel' buttons.

The **GetDigits Module** can be used to accept DTMF (touch tone) input from callers, and optionally play audio files such as IVR menu options to prompt the caller for input.

Module Name: Create PIN

Target Variable: InPIN

Input Format:

Maximum Number of Digits: 4

Termination Digit: #

Max Silence (secs): 2

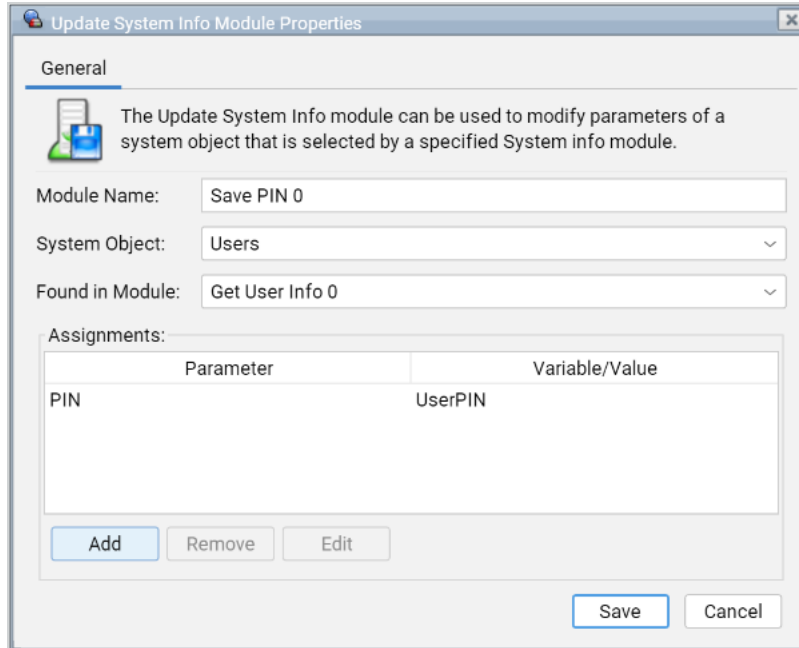
Max Time to Enter Digits (secs): 20

☒ Clear User Input

☐ Collapsible

Update System Info Module

Use this module to save the PIN configured in the previous step to the user account associated with the user's extension.



The Update System Info module can be used to modify parameters of a system object that is selected by a specified System info module.

Module Name: Save PIN 0

System Object: Users

Found in Module: Get User Info 0

Assignments:

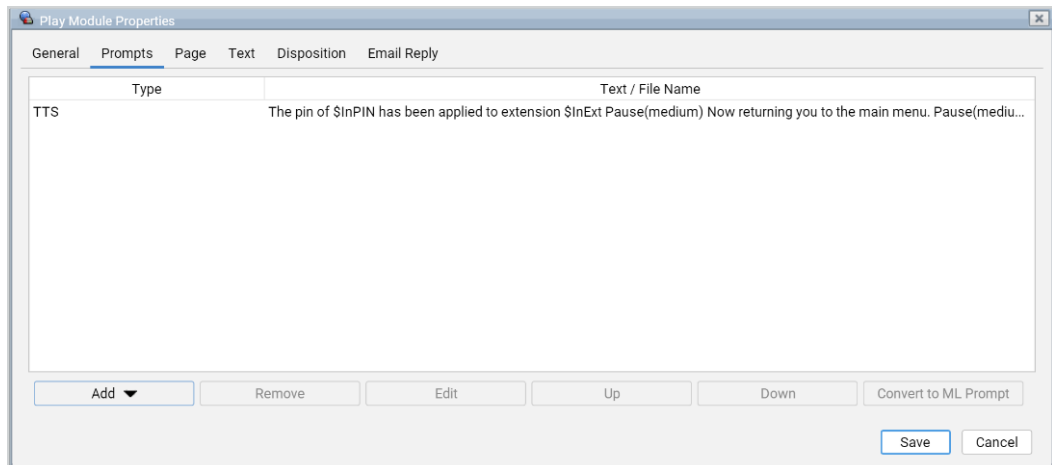
Parameter	Variable/Value
PIN	UserPIN

Add Remove Edit

Save Cancel

Play Module

Use this Play module to provide confirmation for the new PIN. In this module, the `InPIN` and `InExt` variables are used to provide accurate details.



General Prompts Page Text Disposition Email Reply

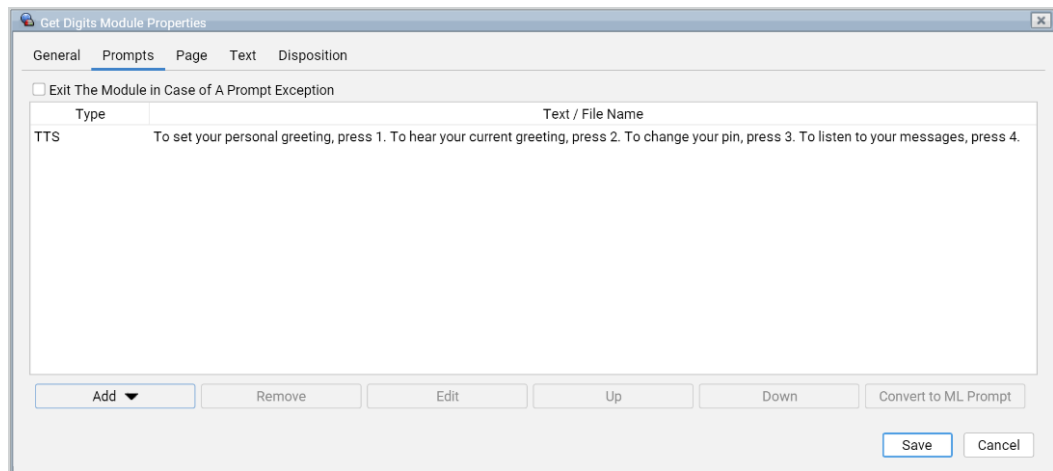
Type	Text / File Name
TTS	The pin of \$InPIN has been applied to extension \$InExt Pause(medium) Now returning you to the main menu. Pause(medium...

Add ▼ Remove Edit Up Down Convert to ML Prompt

Save Cancel

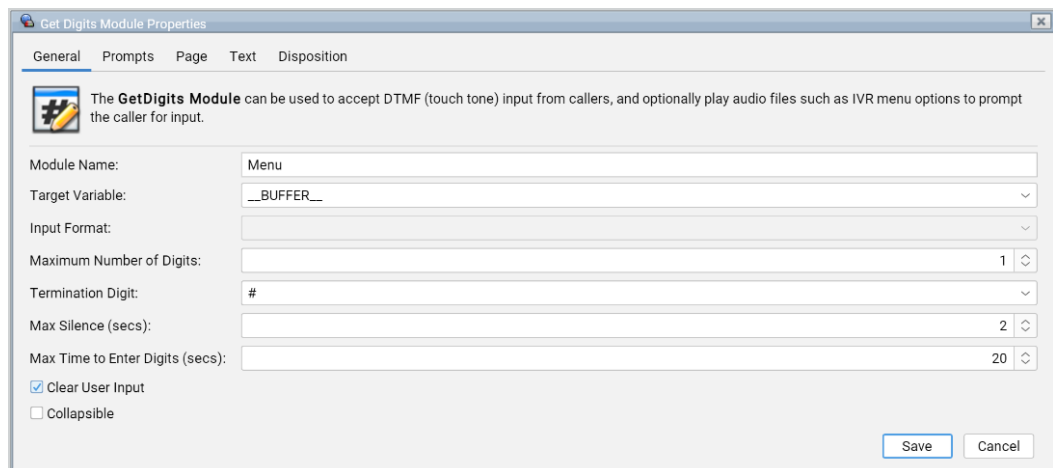
Get Digits Module

Use another Get Digits Module to present menu options to your caller. One option offered is to create a new voicemail greeting. This option is an obvious next step for new user configuration, but may also be used by existing users that want to change their current voicemail greeting.



The screenshot shows the 'Get Digits Module Properties' dialog box with the 'Prompts' tab selected. The 'General' tab is also visible. The 'Exit The Module in Case of A Prompt Exception' checkbox is unchecked. Below it, there is a table with two columns: 'Type' and 'Text / File Name'. The table contains one row with 'TTS' in the 'Type' column and 'To set your personal greeting, press 1. To hear your current greeting, press 2. To change your pin, press 3. To listen to your messages, press 4.' in the 'Text / File Name' column. At the bottom of the dialog, there are buttons for 'Add', 'Remove', 'Edit', 'Up', 'Down', 'Convert to ML Prompt', 'Save', and 'Cancel'.

This module allows a maximum number of one digit to be used input and is configured to allow the caller up to 20 seconds to provide that menu selection choice.



The screenshot shows the 'Get Digits Module Properties' dialog box with the 'General' tab selected. The 'Prompts' tab is also visible. The 'Module Name' is set to 'Menu'. The 'Target Variable' is set to '__BUFFER__'. The 'Input Format' is set to 'DTMF'. The 'Maximum Number of Digits' is set to 1. The 'Termination Digit' is set to '#'. The 'Max Silence (secs)' is set to 2. The 'Max Time to Enter Digits (secs)' is set to 20. The 'Clear User Input' checkbox is checked. The 'Collapsible' checkbox is unchecked. At the bottom of the dialog, there are buttons for 'Save' and 'Cancel'.

Play Module

Use this module to confirm the recorded greeting to the caller.

Get Digits Module Properties

General Prompts Page Text Disposition

The **GetDigits Module** can be used to accept DTMF (touch tone) input from callers, and optionally play audio files such as IVR menu options to prompt the caller for input.

Module Name: Menu

Target Variable: __BUFFER__

Input Format:

Maximum Number of Digits: 1

Termination Digit: #

Max Silence (secs): 2

Max Time to Enter Digits (secs): 20

☒ Clear User Input

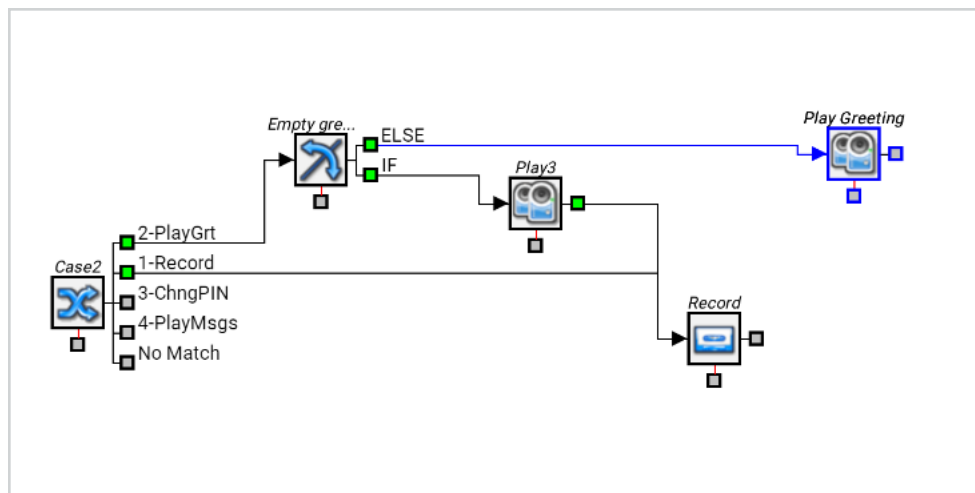
☐ Collapsible

Save Cancel

Connect the output of this module to the input of the main menu options Case module (see [Case Module](#)).

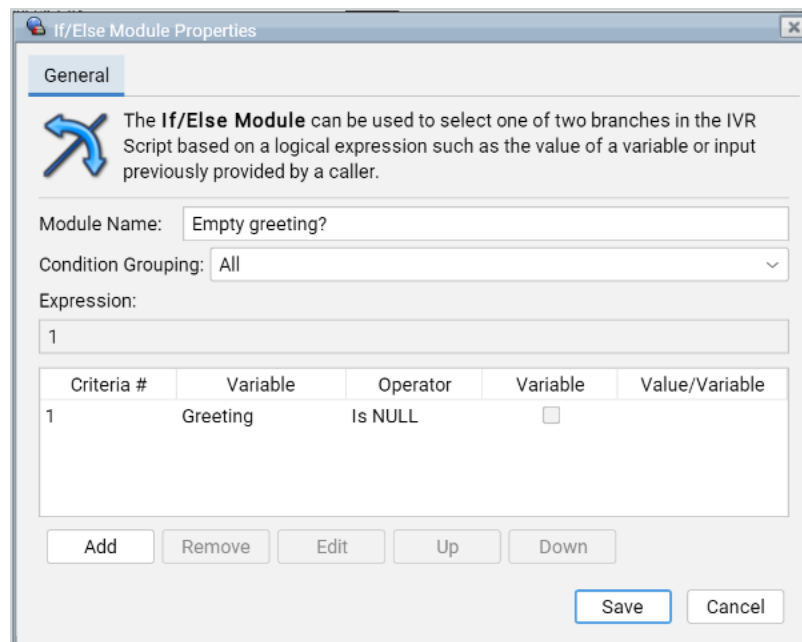
Playing the Current Voicemail Greeting

Another option from the main menu options case module enables the caller to listen to their voicemail greeting. In this branch, an If/Else module first checks the greeting variable for a value.



If/Else Module

Use this module to check the greeting variable to check for the presence of an existing voicemail greeting. A null value indicates no greeting and sends the caller to the Recording a Voicemail Greeting (see [Recording a Voicemail Greeting](#)) branch.

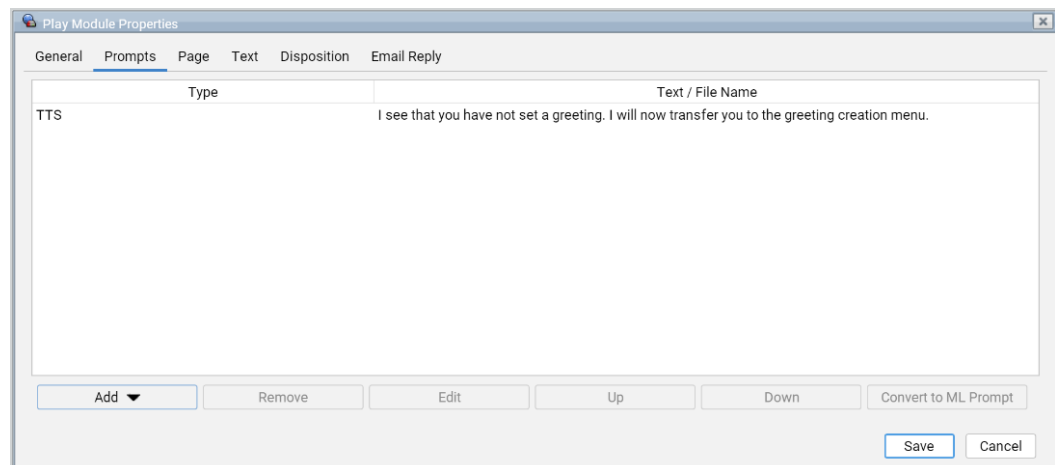


The If/Else Module Properties dialog box is shown. The General tab is active. It contains a description of the module's function, a text field for the Module Name (set to "Empty greeting?"), a dropdown for Condition Grouping (set to "All"), and an Expression field (set to "1"). Below these is a table with columns: Criteria #, Variable, Operator, Variable, and Value/Variable. The table contains one row: 1, Greeting, Is NULL, and an empty checkbox. At the bottom are buttons for Add, Remove, Edit, Up, Down, Save, and Cancel.

Criteria #	Variable	Operator	Variable	Value/Variable
1	Greeting	Is NULL		☐

Play Module

Use this module to inform the caller that they are being redirected to create a voicemail greeting.

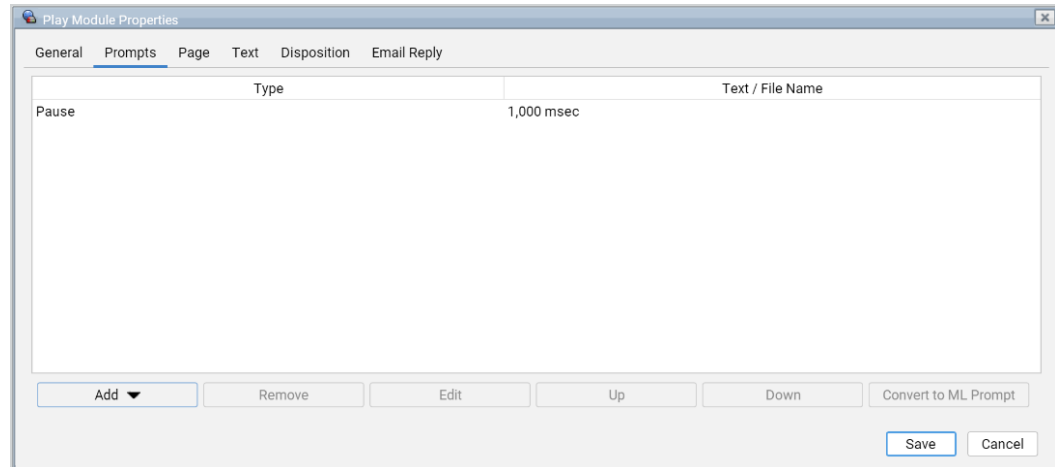


The Play Module Properties dialog box is shown. The Prompts tab is active. It contains a table with columns: Type and Text / File Name. The table has one row: TTS and "I see that you have not set a greeting. I will now transfer you to the greeting creation menu." At the bottom are buttons for Add, Remove, Edit, Up, Down, Convert to ML Prompt, Save, and Cancel.

Type	Text / File Name
TTS	I see that you have not set a greeting. I will now transfer you to the greeting creation menu.

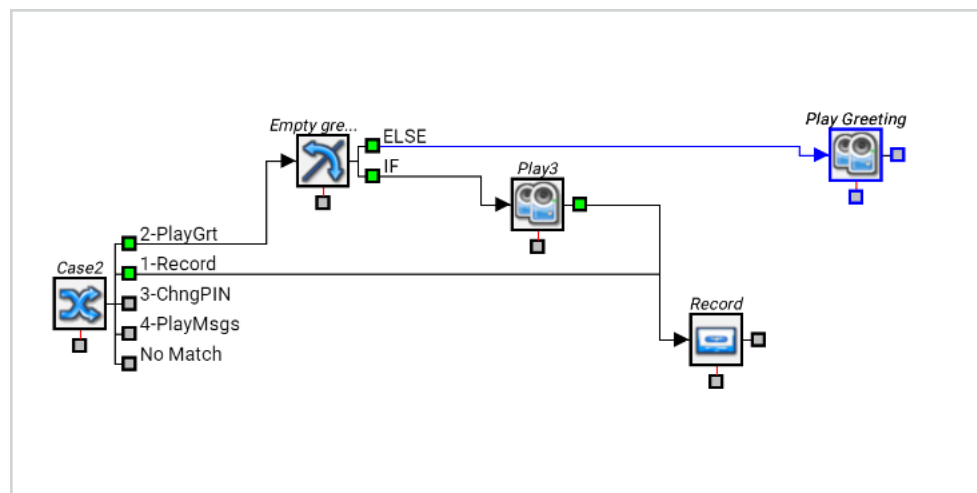
Play Greeting Module

Use this module to retrieve the Greeting variable and play the current voicemail greeting for this caller.



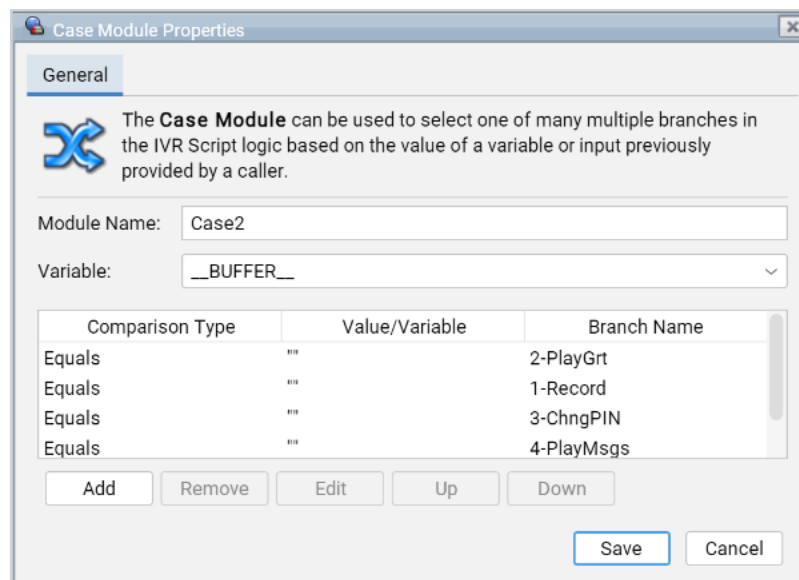
Recording a Voicemail Greeting

This branch of the script enables the recording and storage of a new voicemail greeting. This may be used for new voicemail configuration or for existing users that want to update their current greeting.



Case Module

Use this module as your main menu selection. Configure this module with menu selection options that direct the call flow through the caller's selection. For this branch of the script we will focus only on the record new greeting option. Other options will be detailed in the subsequent branches. This module stores the caller input in the buffer.



The Case Module Properties dialog box is shown with the General tab selected. It contains a description of the Case Module, a text field for the Module Name (Case2), a dropdown for the Variable (__BUFFER__), and a table with four rows of comparison logic. Below the table are buttons for Add, Remove, Edit, Up, and Down. At the bottom right are Save and Cancel buttons.

General

The **Case Module** can be used to select one of many multiple branches in the IVR Script logic based on the value of a variable or input previously provided by a caller.

Module Name: Case2

Variable: __BUFFER__

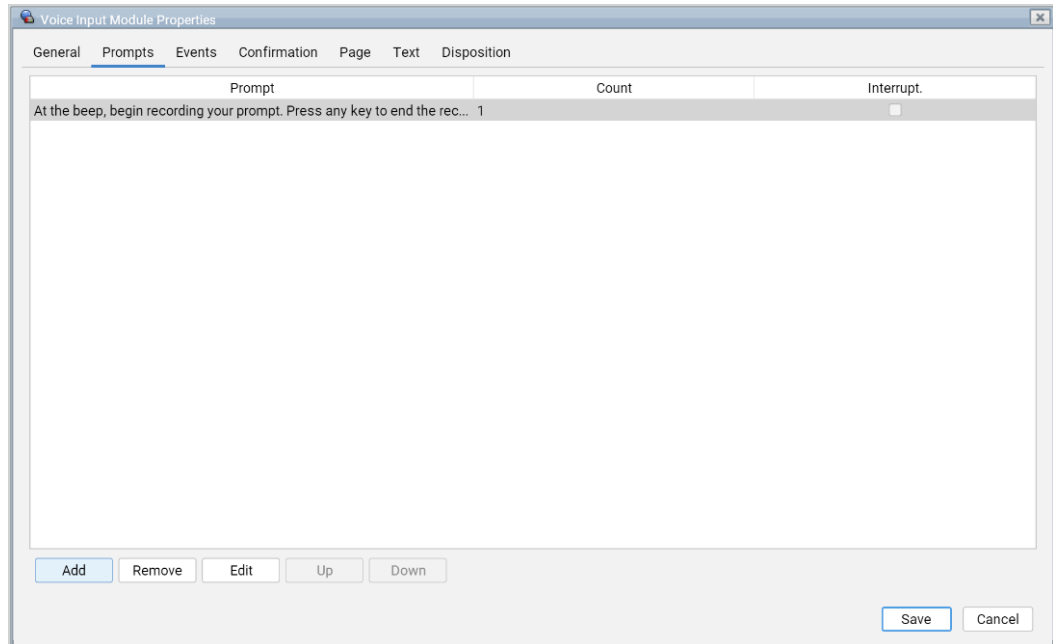
Comparison Type	Value/Variable	Branch Name
Equals	111	2-PlayGrt
Equals	111	1-Record
Equals	111	3-ChngPIN
Equals	111	4-PlayMsgs

Add Remove Edit Up Down

Save Cancel

Voice Input Module

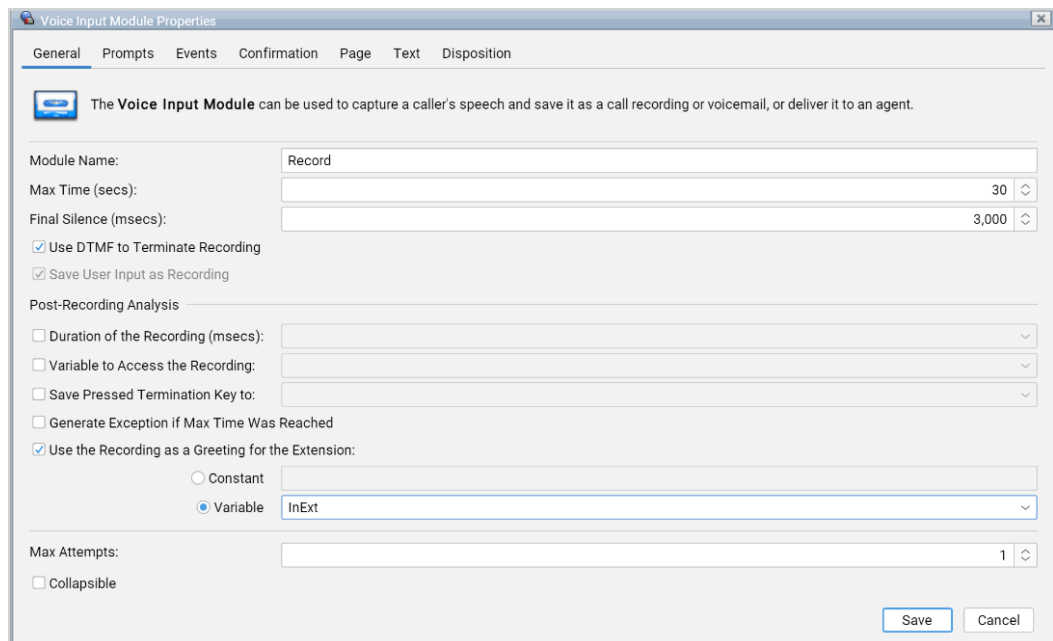
Use this module to prompt the caller to record a voicemail greeting.



The screenshot shows the 'Voice Input Module Properties' dialog box with the 'Prompts' tab selected. The dialog has a title bar with a close button. Below the title bar are tabs: 'General', 'Prompts', 'Events', 'Confirmation', 'Page', 'Text', and 'Disposition'. The 'Prompts' tab contains a table with three columns: 'Prompt', 'Count', and 'Interrupt'. The first row has the prompt 'At the beep, begin recording your prompt. Press any key to end the rec...', a count of '1', and an 'Interrupt' checkbox that is unchecked. Below the table are buttons: 'Add', 'Remove', 'Edit', 'Up', and 'Down'. At the bottom right are 'Save' and 'Cancel' buttons.

Prompt	Count	Interrupt
At the beep, begin recording your prompt. Press any key to end the rec...	1	☐

Configure this module to store the recording as a greeting for the user's extension.



The screenshot shows the 'Voice Input Module Properties' dialog box with the 'General' tab selected. The dialog has a title bar with a close button. Below the title bar are tabs: 'General', 'Prompts', 'Events', 'Confirmation', 'Page', 'Text', and 'Disposition'. The 'General' tab contains a description: 'The Voice Input Module can be used to capture a caller's speech and save it as a call recording or voicemail, or deliver it to an agent.' Below this are fields for 'Module Name' (set to 'Record'), 'Max Time (secs)' (set to 30), and 'Final Silence (msecs)' (set to 3,000). There are checkboxes for 'Use DTMF to Terminate Recording' (checked), 'Save User Input as Recording' (checked), and 'Post-Recording Analysis'. Under 'Post-Recording Analysis', there are checkboxes for 'Duration of the Recording (msecs)', 'Variable to Access the Recording', 'Save Pressed Termination Key to', and 'Generate Exception if Max Time Was Reached'. The 'Use the Recording as a Greeting for the Extension' checkbox is checked, and it has two options: 'Constant' (unchecked) and 'Variable' (checked). The 'Variable' option has a dropdown menu set to 'InExt'. At the bottom are fields for 'Max Attempts' (set to 1) and a 'Collapsible' checkbox. At the bottom right are 'Save' and 'Cancel' buttons.

The **Voice Input Module** can be used to capture a caller's speech and save it as a call recording or voicemail, or deliver it to an agent.

Module Name:

Max Time (secs):

Final Silence (msecs):

☒ Use DTMF to Terminate Recording

☒ Save User Input as Recording

Post-Recording Analysis

☐ Duration of the Recording (msecs):

☐ Variable to Access the Recording:

☐ Save Pressed Termination Key to:

☐ Generate Exception if Max Time Was Reached

☒ Use the Recording as a Greeting for the Extension:

☐ Constant

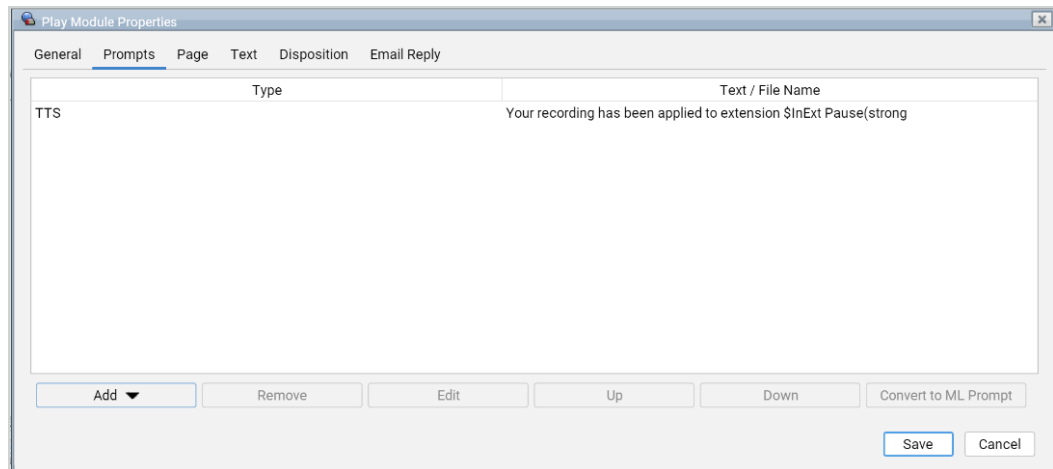
☒ Variable

Max Attempts:

☐ Collapsible

Play Module

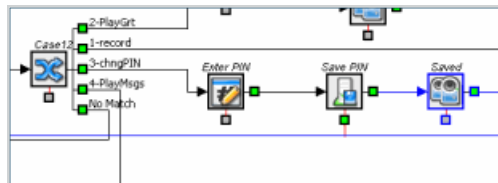
A final Play module confirms to the caller that the new greeting has been applied to their extension.



Connect the output of this module to the input of the main menu options Case module (see [Case Module](#)).

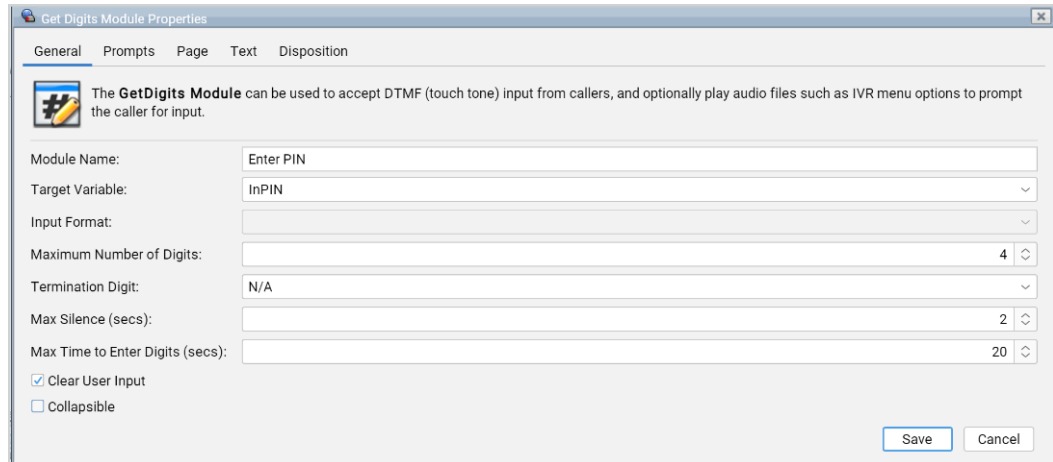
Changing an Existing Voicemail PIN

Use this branch to enable users to establish a new PIN for their voicemail account.



Get Digits Module

Configure this module to allow entry of a four-digit PIN to store in the InPIN variable. Connect the output of this module to the input of an Update System Info module to store the new PIN.



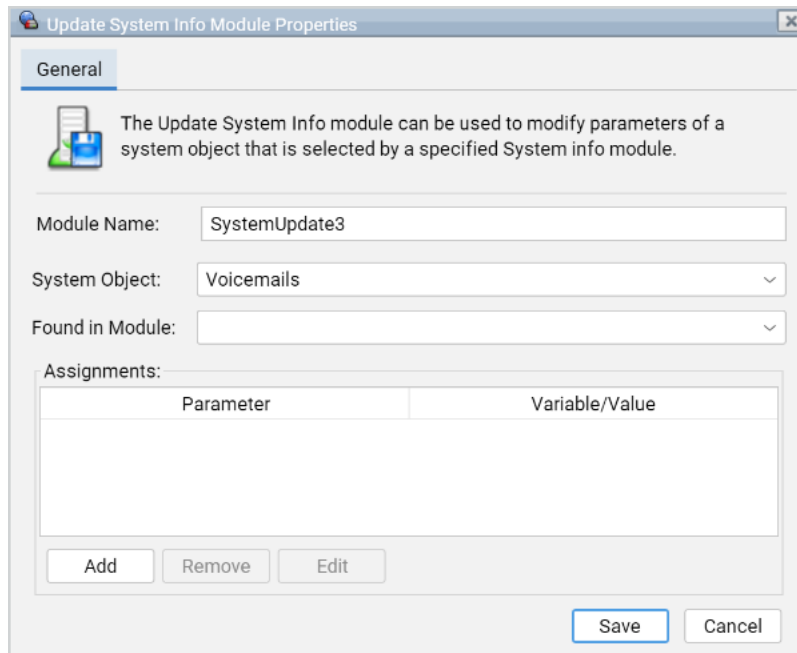
The **Get Digits Module Properties** dialog box is shown with the **General** tab selected. It contains the following fields and options:

- Module Name:** Enter PIN
- Target Variable:** InPIN
- Input Format:** (empty)
- Maximum Number of Digits:** 4
- Termination Digit:** N/A
- Max Silence (secs):** 2
- Max Time to Enter Digits (secs):** 20
- ☒ Clear User Input
- ☐ Collapsible

Buttons: Save, Cancel

Update System Info Module

Configure this module to store the new PIN established in the previous module in the InPIN variable.



The **Update System Info Module Properties** dialog box is shown with the **General** tab selected. It contains the following fields and options:

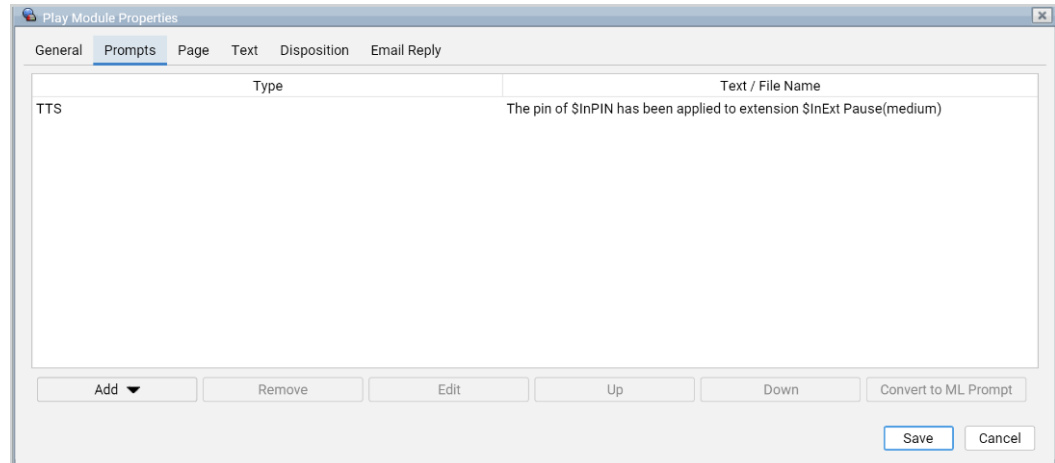
- Module Name:** SystemUpdate3
- System Object:** Voicemails
- Found in Module:** (empty)
- Assignments:**

Parameter	Variable/Value
-----------	----------------

Buttons: Add, Remove, Edit, Save, Cancel

Play Module

Configure this module to provide verbal confirmation that the new PIN has been saved to the caller's extension.



Connect the output of this module to the input of the main menu options Case module (see [Case Module](#)).

Creating an Opt-Out IVR Script

This example creates a IVR script that allows call recipients to opt out of telemarketing calls. It only applies to outbound campaigns.

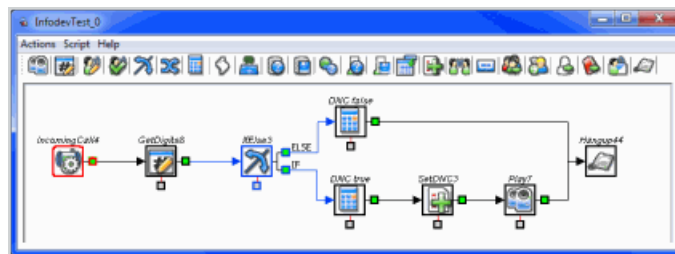
FCC regulations require that all call recipients can add their dialed phone number to the Do Not Call list (DNC) for all abandoned calls during a recorded telemarketing message. When a person answers the phone, a call is considered abandoned or dropped if an agent is not available within 2 seconds of the completed greeting.

Note: The abandon rate 2-second rule calculation option allows you to use a queue time of 2 seconds or higher to identify outbound calls as abandoned. The **Outbound Campaign Abandon Rate % 2 Seconds Rule** standard report shows these values.

According to FCC guideline FCC 03-153, the message played must provide this information:

- Caller: business, Entity, or individual on whose behalf the call is made.
- Purpose: Telemarketing
- Phone number: To enable call recipients to place their number on the DNC list.
- Opt-out process: Phone key or IVR option that enables call recipients to add their number automatically to the DNC list before ending the call.

This example describes how to create this IVR script:



[Creating a Variable to Track Calls and Numbers Added to the DNC List](#)

[Creating the IVR Script](#)

[Configuring the Outbound Campaign Properties](#)

[Testing Your IVR Script and Outbound Campaign](#)

Creating a Variable to Track Calls and Numbers Added to the DNC List

This variable may contain either a boolean or string value. For example, in your Call Variables folder, create the boolean variable isDNC. When true, the variable tracks the DNIS added to the DNC list.

To include the variable in call log reports, check **Reporting Call Variable (Saved in Database)**. Under the Reporting tab, associate the variable with the disposition named Abandon.

The screenshot shows a window titled "New Call Variable Properties" with two tabs: "General" and "Reporting". The "Reporting" tab is selected. Inside the "Reporting" tab, there is a text input field containing "isDNC". Below this is a "Description:" label followed by an empty text input field. There are two checkboxes: "Reporting Call Variable (Saved in Database)" which is checked, and "Field Contains Sensitive Data" which is unchecked. Below the checkboxes is a "Data Type:" label followed by a dropdown menu showing "Boolean". At the bottom of the dialog, there are three tabs: "Restrictions", "List Items", and "Default Value". The "Restrictions" tab is selected. At the bottom right of the dialog are "Save" and "Cancel" buttons.

Creating the IVR Script

The IVR script that you link to the outbound campaign must contain these elements:

- [Get Digits Module](#)
- [If/Else Module](#)
- [Set Variable Modules](#)
- [Set DNC Module](#)
- [Play Module](#)
- [Hangup Modules](#)

Get Digits Module

This module routes calls according to the digits that it receives. In this case, a single digit is sufficient.

- 1 In the General tab, set the Maximum Number of Digits to 1.

The screenshot shows the 'Get Digits Module Properties' dialog box with the 'General' tab selected. The dialog box has a title bar and a close button. Below the title bar are tabs for 'General', 'Prompts', 'Page', 'Text', and 'Disposition'. The 'General' tab contains a description of the module, a list of properties, and 'Save' and 'Cancel' buttons at the bottom right.

Get Digits Module Properties

The **GetDigits Module** can be used to accept DTMF (touch tone) input from callers, and optionally play audio files such as IVR menu options to prompt the caller for input.

Module Name: GetDigits6

Target Variable: __BUFFER__

Input Format:

Maximum Number of Digits: 1

Termination Digit: #

Max Silence (secs): 2

Max Time to Enter Digits (secs): 20

☒ Clear User Input

☐ Collapsible

Save Cancel

- 2 In the Prompts tab, add a prompt to describe the call to recipients and the option to add their number to the DNC list.

For example, you can use a recorded prompt or a TTS prompt with the following text:

This is ABC Company calling for telemarketing purposes. Our phone number is 1-800-555-1212. If you would like to be added to our Do Not Call list, please press 9 now.

The screenshot shows the 'Get Digits Module Properties' dialog box with the 'Prompts' tab selected. The dialog box has a title bar and a close button. Below the title bar are tabs for 'General', 'Prompts', 'Page', 'Text', and 'Disposition'. The 'Prompts' tab contains a checkbox for 'Exit The Module in Case of A Prompt Exception', a table with 'Type' and 'Text / File Name' columns, and 'Add', 'Remove', 'Edit', 'Up', 'Down', and 'Convert to ML Prompt' buttons at the bottom. 'Save' and 'Cancel' buttons are at the bottom right.

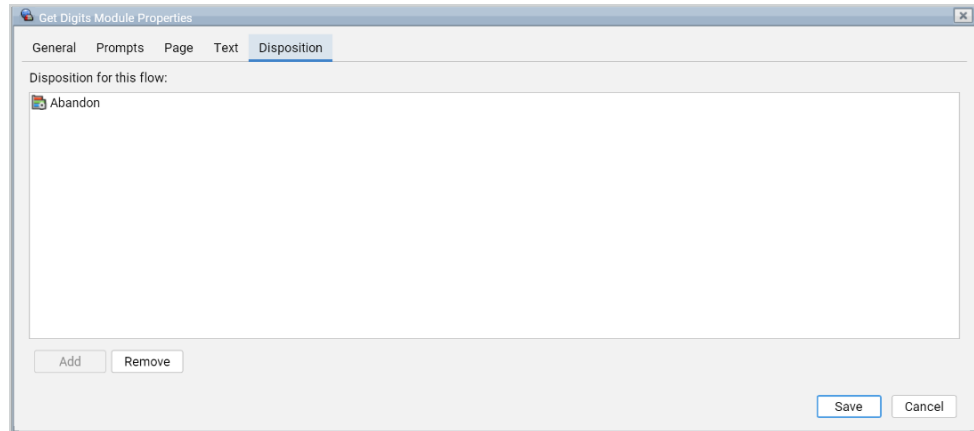
☐ Exit The Module in Case of A Prompt Exception

Type	Text / File Name
TTS	This is ABC Company calling for telemarketing purposes. Our phone number is 1-800-555-1212. If you would like to be added to our Do Not Call list, plea...

Add Remove Edit Up Down Convert to ML Prompt

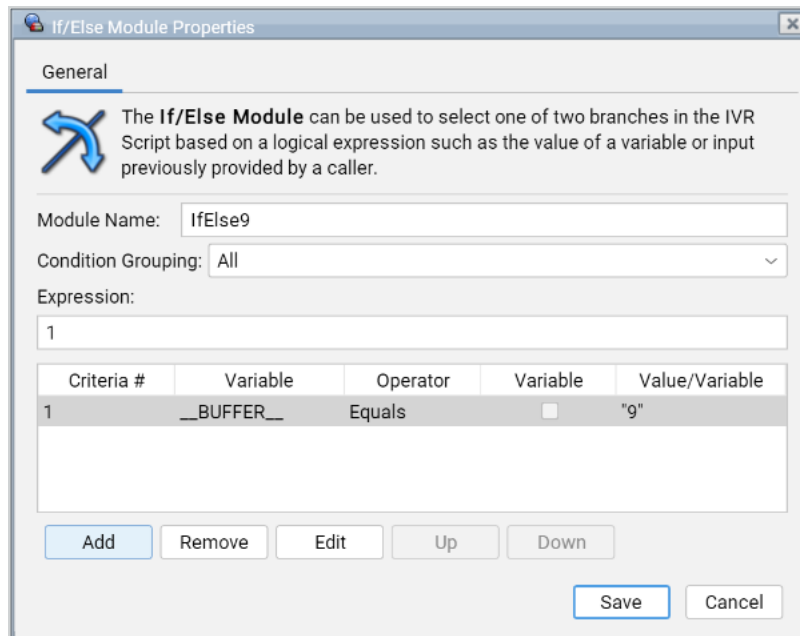
Save Cancel

- 3 On the Disposition tab, set the Disposition to Abandon.



If/Else Module

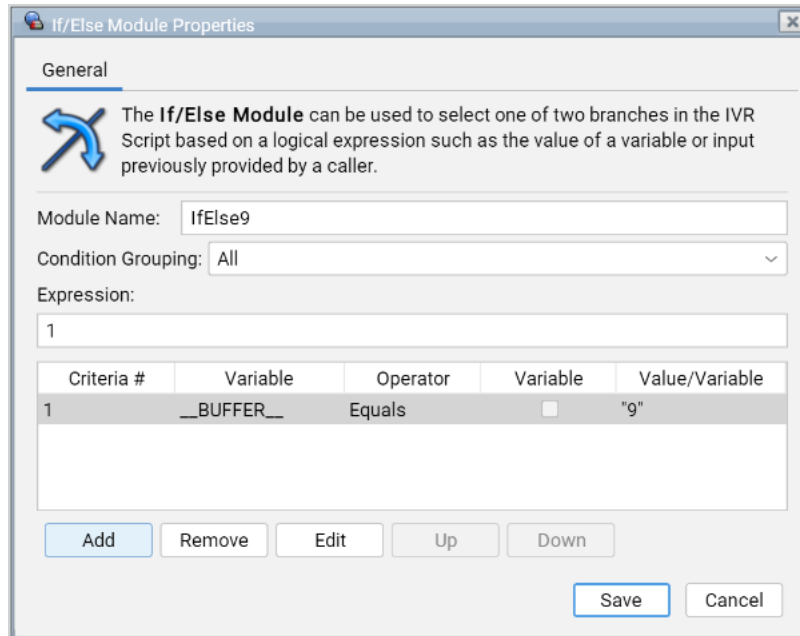
This module processes the response from the call recipient. Configure the if (true) branch as follows: `BUFFER = 9`. Later you will use the Set DNC module to add the called number to the DNC list. In all other cases, the number is not added to the DNC list.



Set Variable Modules

Connect the If/Else branches to the Set Variable module as follows:

- If branch to DNCTrue
- Else branch to DNCFalse



The **If/Else Module** can be used to select one of two branches in the IVR Script based on a logical expression such as the value of a variable or input previously provided by a caller.

Module Name:

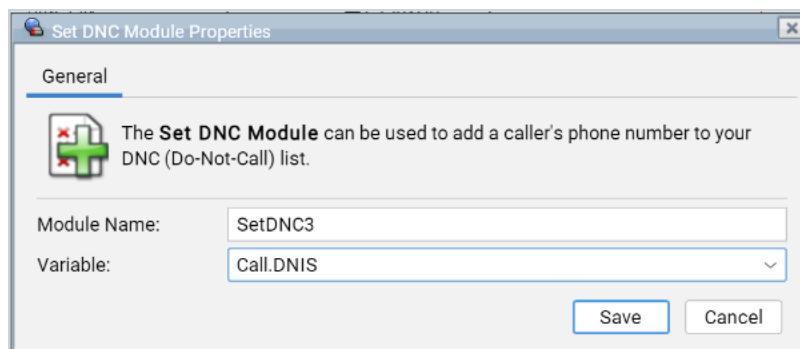
Condition Grouping:

Expression:

Criteria #	Variable	Operator	Variable	Value/Variable
1	__BUFFER__	Equals	☐	"9"

Set DNC Module

Add the Set DNC module to the If branch and use the Call.DNIS variable value.



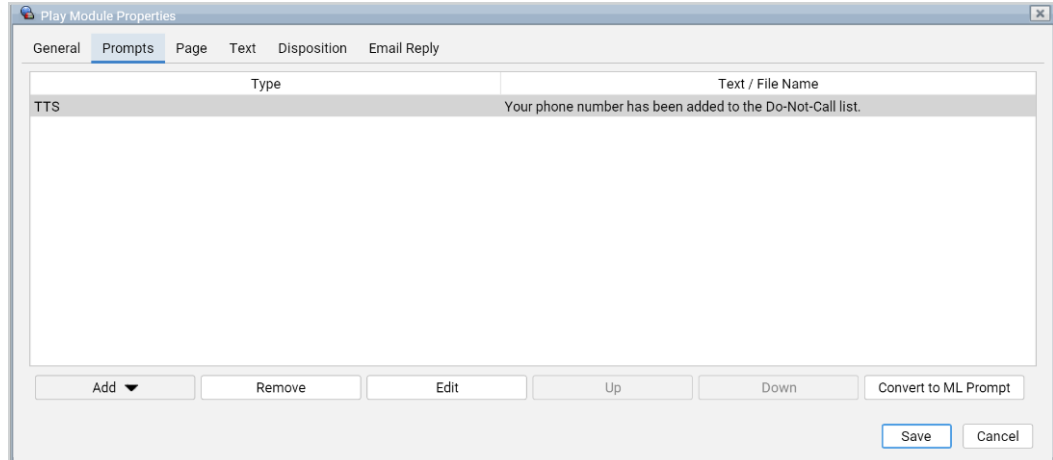
The **Set DNC Module** can be used to add a caller's phone number to your DNC (Do-Not-Call) list.

Module Name:

Variable:

Play Module

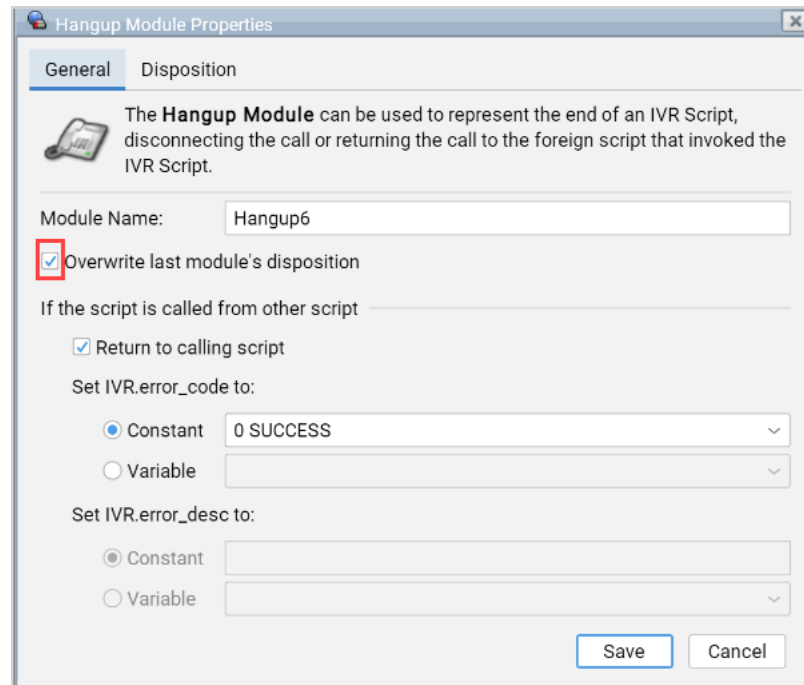
Use this module to confirm to call recipients that you placed their phone number on the DNC list. Set the disposition to Abandon.



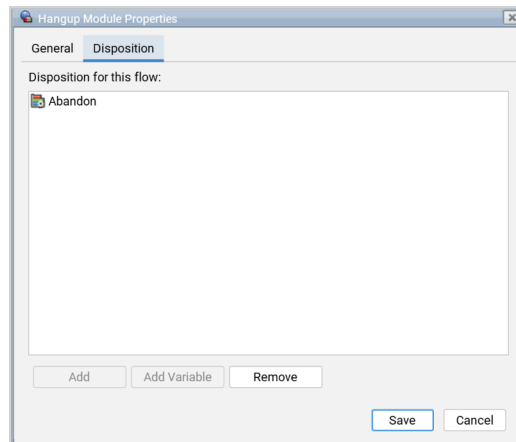
Hangup Modules

Configure the Hangup module in the main workspace and the module in the lower pane in the same manner. This last module is used if the call is dropped before the Get Digits module.

- 1 In the General tab, check **Overwrite last module's disposition** to activate the **Disposition** tab.



- 2 In the Disposition tab, set the disposition to **Abandon**.



Configuring the Outbound Campaign Properties

The outbound campaign properties must comply with FCC regulations.

DC Properties

Call Survey Lists Prompts Skills Dispositions Connectors Worksheet Script E-Mail

General Recording Transcripts **Dialing Options** Time Zone Dialing

Campaign Dialing Mode

☐ Predictive ☒ Power ☐ Progressive ☐ Preview (Use Progressive when blending with inbound campaigns for less than 15 agents)

List Dialing Mode

☐ Vertical Dialing ☒ List Penetration ☐ Extended Strategy [Manage Strategy](#)

Campaign Priority: 3 Campaign Ratio: 50

Note: To enable Priority and Ratio, please turn on corresponding setting in the Domain configuration

Minimum Duration Before Redialing Number (HH:MM): 00:30

Call(s) To Agent Ratio

Abandoned Call Percentage

☐ Monitor Abandoned Call Percentage

Max Abandon Call Percentage

Call Analysis

☐ No Call Analysis ☐ Fax Detection Only ☒ Fax and Answering Machine

Voice Detection Level

Fast - 2 sec Normal - 4.5 sec Accurate - 10 sec

Action on Answering Machine Detection: Drop Call

Action on Max Queue Time Expiration: Abandon Call

Apply **Save** Cancel

In the General settings tab, set the Max Queue Time to 1 second. In the Dialing Options tab, set these options:

- **Dialing Mode:** Add the opt-out feature to these campaign dialing modes:
 - Predictive
 - Power
 - Progressive

Note

The preview mode gives the call to the agent first so the agent can add the caller to the DNC list.

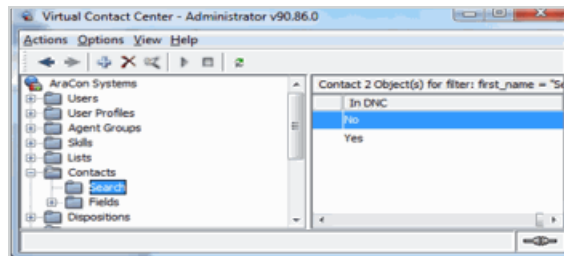
- **Dropped Call Percentage:** Check **Monitor Dropped Call Percentage** and set the slider appropriately to comply with FCC regulations, for example: 3%.

- **Action on Max Queue Timeout Expiration:** Select the IVR script that you created in the previous section.

Testing Your IVR Script and Outbound Campaign

To test your outbound campaign, at least one agent with the correct skill must be logged in with the status **Ready for Call**. If no agent is logged in, calls are not dialed.

- 1 Start your campaign.
All numbers in the list are dialed, including a number that will reach you.
- 2 Listen to the prompt, press 9, and listen the confirmation message.
The number is added to the DNC list.
- 3 Restart the campaign.
- 4 Right-click and select **Reset List Position** or another resetting actions.
The number assigned to the DNC is not called.
- 5 Validate the Results.
 - In the **Contacts > Search** folder, find the contact. In the In DNC column, the value is **Yes** as shown below:

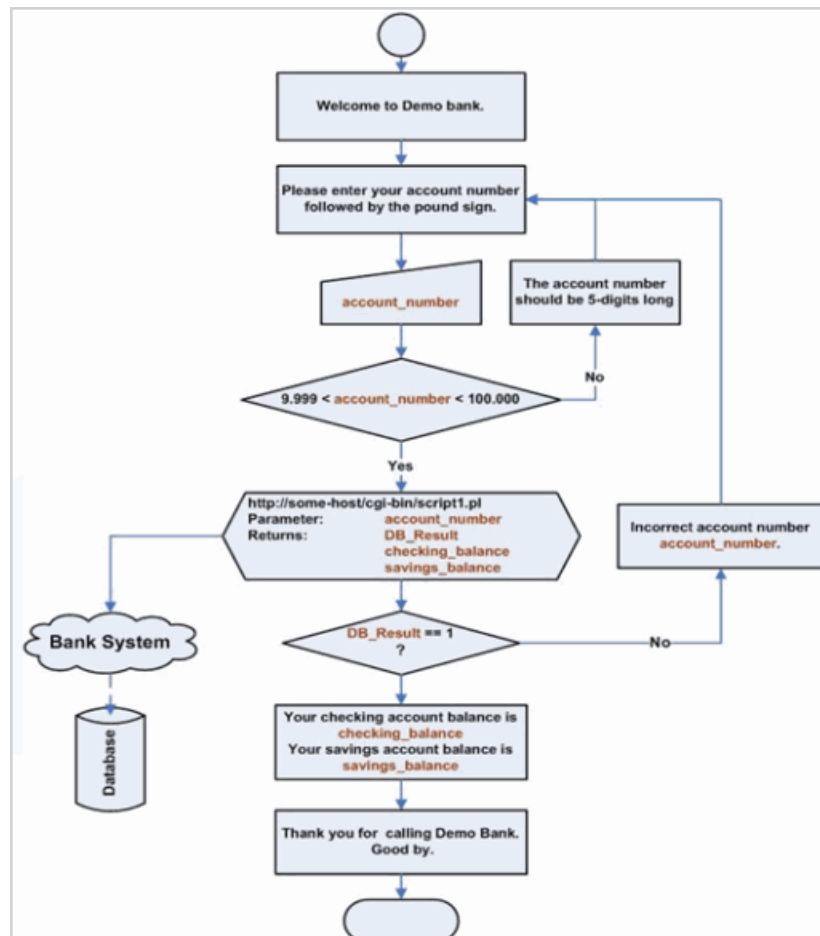


- The variable appears in the Call Log report. The disposition value for the call is Abandon as shown below:

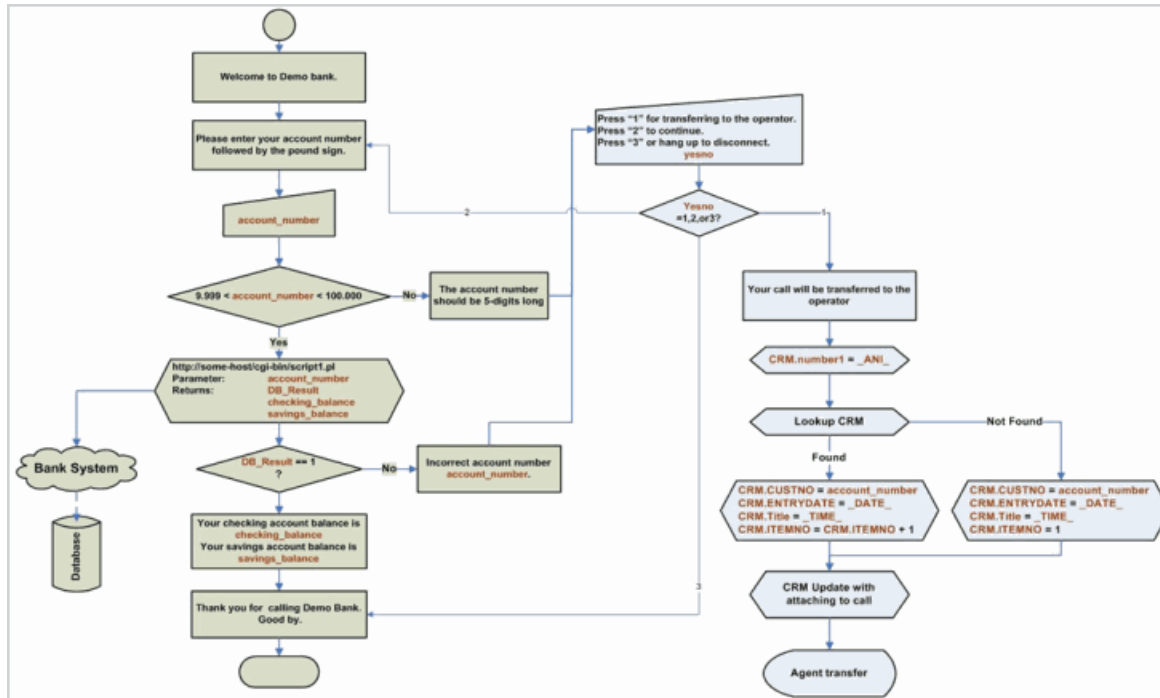
CALL ID	TIMESTAMP	CAMPAIGN	CALL TYPE	AGENT	AGENT NAME	DISPOSITION	Opt-out-to-DNC	ANI	CUSTOMER NAME
187871	Wed, 10 Jan 2013 08:01:17	[None]	Station	adept1_ar	adept1_ar	Station Session	0	9252345671	-
187872	Wed, 10 Jan 2013 08:36:10	preclaim_in	Inbound	[None]	-	Abandon	0	8314401886	-
187873	Wed, 10 Jan 2013 13:53:41	gpheller_outbound_opt_out	Outbound	[None]	-	Abandon	1	9252345671	Ruff at Desk

Flowcharts

This section contains flowcharts for complex IVR scripts. Before creating an IVR, always map your script in this manner.



More complex example:



Error Messages

This table contains the errors that can occur for each module.

Code	Condition	Description	Modules
		No records selected in the SysInfoModule.	Update System Info
		System Object and Found in Module are incompatible.	Update System Info
		Found in Module is not populated.	Update System Info
	INCORRECT_DATA	Variables contain incorrect values.	Update System Info
1001	SYSTEM_ERR	System error	
1002	TTS_ERROR	TTS error	
1003	TTS_OUT_OF_LICENSE	TTS or ASR out of licenses	
1050	INVALID_USER_INPUT	Invalid user input	Get Digits
1051	NO_USER_INPUT	No user input	Get Digits
1100	VARIABLE_ERROR	Incorrect operation with variable	Set Variable
1101	INVALID_RESPONSE	Invalid response from server	Query
1102	MALFORMED_URL	Malformed URL	Query
1103	FAILED_TO_CONNECT	Failed to connect to server	Query
1104	REQUEST_TOO_BIG	Size of request to Web server exceeds limit	Query
1105	AUTH_PROFILE_NOT_FOUND	Auth profile does not exist	Query
1106	JS_EXECUTION_TIMEOUT	JS function execution timeout	Query
1107	SECURE_VARS_IN_URL_ENCODED_PARAMS	Secure variables are used in URL encoded parameters	Query
1108	SECURE_VARS_IN_	HTTP requests cannot use secure	Query

Code	Condition	Description	Modules
	HTTP_REQUEST	variables	
1150	CANT_ASSIGN_ARGUMENTS	Cannot assign arguments while calling foreign script	Foreign Script
1200	CRM_RECORD_NOT_FOUND	Contact record does not exist	Lookup Contact Record
1201	CRM_NOT_UNIQUE	More than one contact record found	Lookup Contact Record
1202	CRM_IS_NOT_SELECTED	Contact record not selected	Contact Update
1203	CRM_ADVANCED_CRITERIA_TOO_WIDE	Result contains more than 100 records	Lookup Contact Record
1204	DB_REQUEST_TIMEDOUT	Database request is timed out	Lookup Contact Record
1250	CRM_RECORD_ALREADY_EXISTS	Cannot add contact record because it already exists in the database	Contact Update
1251	CRM_FIELD_VALUE_INCORRECT	Incorrect value of contact field	Contact Update
1300	INVALID_VALUE_IN_CONDITION	Failed to calculate condition because of invalid arguments	Case, If/Else
1400	ANSMACH_UNDEFINED	Cannot determine answer machine state	Answering Machine
1401	FAX_DETECTED	Fax detected	Answering Machine
1500	INVALID_NUMBER	Invalid number	Set DNC, Third Party Transfer
1501	NO_CHANNEL_AVAILABLE	No phone line available to make a call	Set DNC, Third Party Transfer
1600	PROMPT_DOESNOT_EXIST	Invalid prompt name	Play
1601	ANSWERMACHINE_DETECTION_IN_PROGRESS	Cannot perform operation while answer machine detection is in progress	Play
1700	INVALID_AGENT_OR_EXTENSION	User with specified agent name or extension was not found, agent inactive, or user does not have agent role.	Agent/Voice Mail Transfer
1701	TRANSFER_TO_	Transfer to campaign failed. Check	Agent/Voicemail Transfer

Code	Condition	Description	Modules
	CAMPAIGN_FAILED	that the campaign is active.	
1800	ITERATOR_INDEX_OUT_OF_RANGE	Iterator index is out of range	Iterator
1801	ARRAY_IS_EMPTY_OR_DOES_NOT_EXIST	Array is empty or does not exist	Iterator
1900	INPUT_NOT_CONFIRMED_OR_NO_INPUT	User has not entered or confirmed the input	Input, Voice Input
1901	ATTEMPTS_MAX_NUMBER_REACHED	Maximum number of attempts is reached	Input, Voice Input
1950	MAX_SIMULTANEOUS_RECORDINGS_REACHED	Maximum number of simultaneous recording reached	Voice Input
2000	NO_CHOICES_TO_CHOOSE	No available choices or referenced data is not initialized	Menu
2100	NO_RECORDS	SystemInfo for given conditions does not exist	System Info
2101	SYSTEM_INFO_NOT_UNIQ	More than one System Info record found	System Info
2102	WEB_ANALYTICS_NOT_ALLOWED	Web analytics is not allowed	System Info
2200	DATA_RESTRICTION_VIOLATION	Data type or restriction violation	Agent/Voicemail Transfer, Answering Machine, Case, Conference, Contact Update, Extension Transfer, Foreign Script, Get Digits, Hangup, If/Else, Input, Iterator, Lookup Contact Record, Menu, Play, Query, Set DNC, Set Variable, Skill transfer, System Info, Third Party Transfer, Voice Input, Voicemail Transfer
2201	INVALID_CAMPAIGN_NAME_OR_EXTENSION	Invalid campaign name or extensions	System Info

Code	Condition	Description	Modules
2300	INVALID_SKILL	Skill does not exist. Appears if the list of skills returned in a System Info module is empty.	Iterator, Skill transfer
2400	INCORRECT_DATA_TO_UPDATE	System update is failed due to incorrect data	System Update
2500	NOT_APPLICABLE_MODULE	The module cannot be executed in current media	Agent/Voicemail Transfer, Conference, Extension Transfer, IVA Transfer, Third Party Transfer, Voicemail Transfer
2600	LANGUAGE_NOT_EXIST	Language does not exist in this script	Language
2601	VARIABLE_NOT_FOUND	Variable is not found	Language
2602	URI_INVALID	Incorrect reference to grammar file	Language
2603	GRAMMAR_INVALID	Incorrect content of grammar file	Language
2604	CRM_IS_OBSOLETE	Contact Record is obsolete	Contact Update
2700	FAILED_TO_GET_DATA_TO_SEND_TO_3RD_PARTY	Failed to get data to send to third party	Third Party Transfer
2701	TIMEOUT_3RD_PARTY_CALL	Call duration on third party side exceeds max timeout	Third Party Transfer
2702	FAILED_MAKE_3RD_PARTY_CALL	Failed to make third party call	Third Party Transfer
2703	FAILED_TRANSMIT_DATA_TO_3RD_PARTY	Failed to transmit data to third party	Third Party Transfer
2704	FAILED_RECEIVE_DATA_FROM_3RD_PARTY	Failed to receive data from third party	Third Party Transfer
2705	THIRD_PARTY_DESCENDANT_NOT_DEFINED	Third party transfer does not have a descendant	Third Party Transfer
2706	RINGING_TIMEOUT	No answer during ringing timeout	Third Party Transfer
2800	FAILED_TO_GET_DATA_TO_SEND_TO_IVA	Failed to get data to send to IVA	IVA Transfer

Code	Condition	Description	Modules
	IVA		
2801	TIMEOUT_IVA_CALL	Call duration on IVA side exceeds max timeout	IVA Transfer
2802	FAILED_MAKE_IVA_CALL	Failed to make IVA call	IVA Transfer
2803	IVA_DESCENDANT_NOT_DEFINED	IVA transfer doesn't have descendant	IVA Transfer
91	CUSTOM_ERR_1		
92	CUSTOM_ERR_2		
93	CUSTOM_ERR_3		
94	CUSTOM_ERR_4		
95	CUSTOM_ERR_5		

AI Agent Connect

AI Agent Connect integrates your Five9 IVR system with certified third-party AI providers. These integrations streamline customer onboarding and support cloud-to-cloud and business-to-business scenarios.

This topic explains how to implement each certified AI provider. The implementation guides for each provider appear in the sections that follow. Before you start, review the prerequisites in the implementation guide for that provider. To complete the setup, get the required configuration details directly from your AI provider.

You can implement only pre-integrated, certified AI providers. If you don't see an integration guide for your provider, contact your sales representative to confirm certification status and availability.

[Cresta.AI Implementation Guide](#)
[Decagon Implementation Guide](#)
[FlipCX Implementation Guide](#)
[Hyro Implementation Guide](#)
[Kastle.AI Implementation Guide](#)
[LevelAI Implementation Guide](#)
[Parloa.AI Implementation Guide](#)
[Sierra Implementation Guide](#)

Cresta.AI Implementation Guide

This section explains integrating Cresta.AI, a certified third-party AI provider, with the Five9 platform.

Prerequisites

Before you begin, make sure you have:

- A Five9-provisioned domain
- A Five9 domain ID
- A Cresta.AI-activated account
- A Cresta.AI-assigned phone number
- Cresta.AI-generated query credentials

Integration Process

The following tasks outline what each party does during the Cresta.AI integration.

What Cresta.AI Does (Informational Only)

- 1 Creates your account in the Cresta.AI platform.
- 2 Configures Five9-provided domain ID.
- 3 Provides the integration details required to begin setup.
- 4 Uses **X-headers** to differentiate routing to its AI platform when multiple AI agents exist.

What You Do (Customer Responsibility)

- 1 Provide your Cresta.AI phone number (one per customer) to the Five9 implementation team.
- 2 Provide your Cresta.AI query credentials, when applicable, including the API username and password for the Query module.

What the Five9 Implementation Team Does (Informational Only)

- 1 Provides your domain ID to complete the IVA account setup.
- 2 Applies your AI phone number for outbound call routing through the Cresta.AI integration.
- 3 Creates a campaign and assigns a test direct inward dialing (DID) number.

- 4 Configures call-attached variables (CAVs) to enable reporting. The delivery team will advise you if additional reporting variables are required.
- 5 Imports the Five9 IVR template using the default configuration. For more information about transfer settings, see [IVR Transfer Configuration](#).
- 6 Applies your AI phone number to the third-party transfer module.
- 7 Tests the campaign to confirm that calls successfully reach Cresta.AI.

IVR Transfer Configuration

The integration sends **X-headers** on outbound invites from the Five9 platform. The platform retrieves data from Cresta.AI using the Query module.

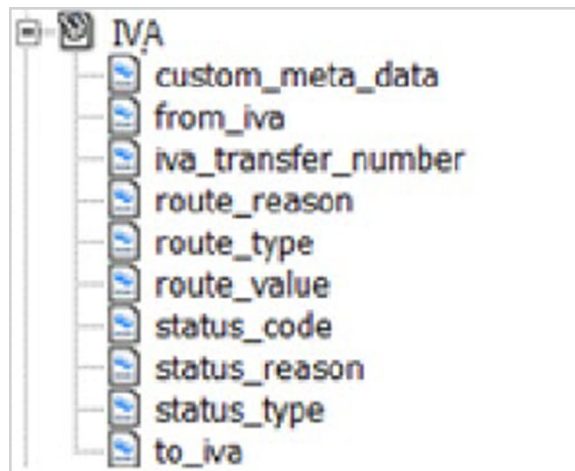
The Five9 implementation team completes the following IVR transfer configuration steps to configure the Five9 platform for call transfers that use default **X-headers**.

Note: As customer, you don't configure the IVR. The Five9 implementation team performs the IVR transfer configuration steps. You can use the following information for awareness only.

- 1 Creates a call variable group named **CrestaAI**.
- 2 Creates the following CAVs:

Variable	Type	Description
custom_meta_data	String	Needed return data (breadcrumb, short summary)
from_iva	String	Data returned from the IVA (KV list)
iva_transfer_number	String	AI phone number
route_reason	String	Short outcome description for Five9 reporting
route_type	String	Next action (skill transfer, phone transfer)
route_value	String	Destination (skill name,

Variable	Type	Description
		phone number)
status_type	String	Event type that triggered status (transfer, API)
status_code	Integer	Numeric error code
status_reason	String	Error reason
to_iva	String	Data sent to the IVA (KV list)



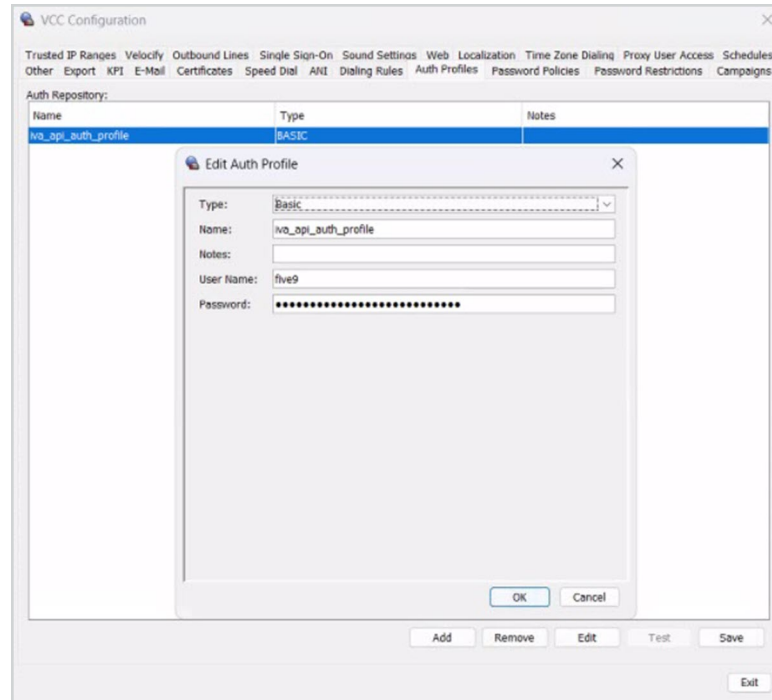
Note: The IVR script automatically populates these CAV values from corresponding **X-headers** in the API. For more information about headers, see [Header Usage](#).

3 Configures the Auth Profile. If the integration requires API use, the Five9 implementation team:

- a** Creates an authentication profile using your Cresta.AI credentials.
- b** Names the profile **Cresta.ai**.

If the integration doesn't require API access, the team skips this step.

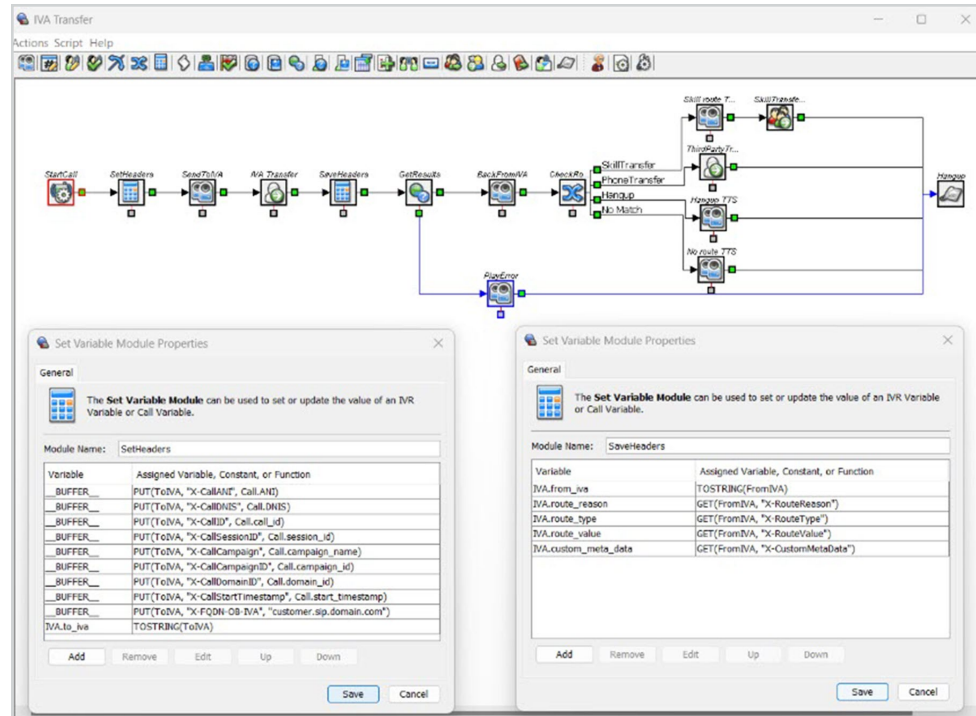
The following example shows an Auth Profile configuration.



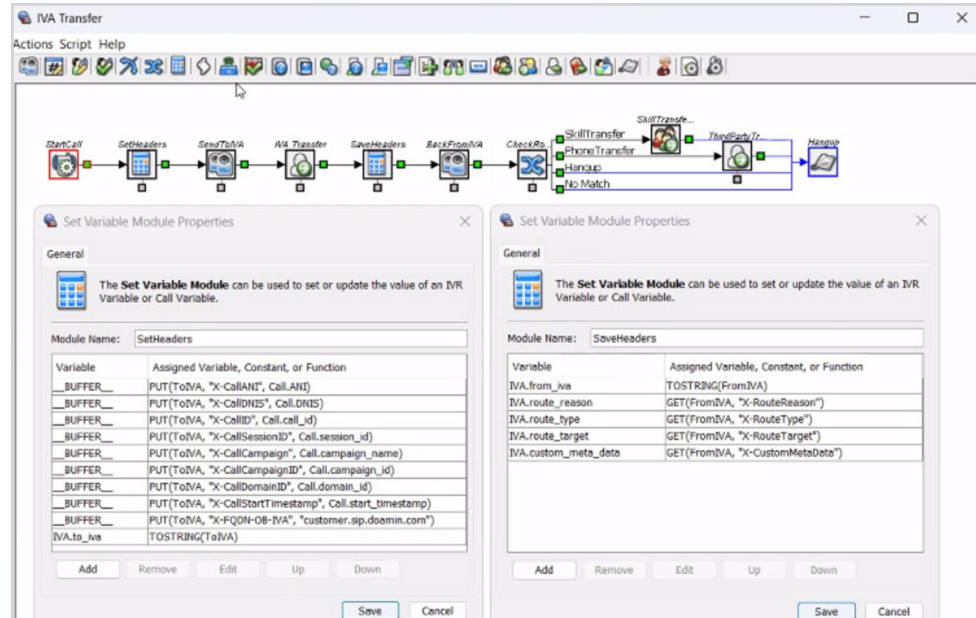
4 Imports the **Cresta AI Agent Connect Integration v1** IVR script.

Note: The Five9 implementation team uses the template designed specifically for Cresta.AI, as **X-header** names vary across integrations.

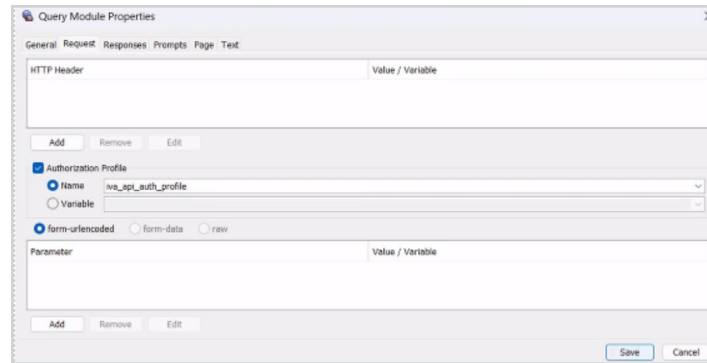
The following example shows an IVR that uses a Query module.



The following example shows an IVR that doesn't use a Query module.

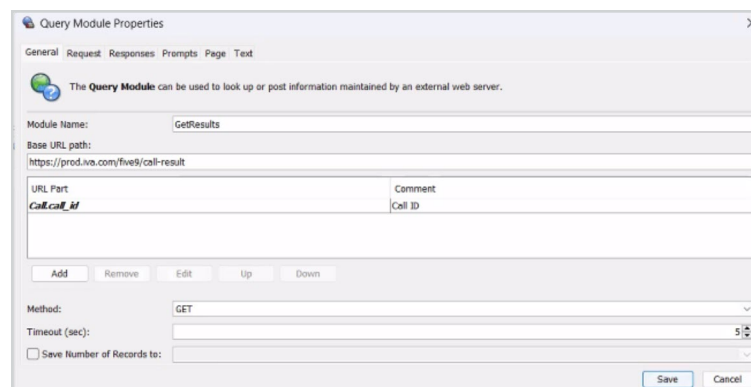


- 5 When applicable, assigns the Cresta.AI auth profile to the Query module. The following example shows an assigned auth profile to the Query module.



The screenshot shows the 'Query Module Properties' dialog box with the 'General' tab selected. The 'Authorization Profile' section is checked, and the 'Name' is set to 'iva_api_auth_profile'. The 'Form-urlencoded' radio button is selected. The 'HTTP Header' and 'Parameter' sections are empty.

The system already populates the production URL in the IVR template. The following example shows a pre-populated production URL in the IVR template.



The screenshot shows the 'Query Module Properties' dialog box with the 'General' tab selected. The 'Module Name' is 'GetResults'. The 'Base URL path' is 'https://prod.iva.com/five9/call-result'. The 'URL Part' is 'call_id'. The 'Method' is 'GET' and the 'Timeout (sec)' is 5. The 'Save Number of Records to' checkbox is unchecked.

- 6 Configures the transfer number by using the single AI phone number that Cresta.AI provides for your organization. This configuration supports one phone number per customer, regardless of the number of domains. The Five9 implementation team sets the default value of the **iva_transfer_number** variable to this phone number.

Note: The Connectivity Project Manager manages call routing for the AI phone number to the IVA integration. The Five9 implementation team also uses the same transfer DID across all your domains that send calls to Cresta.AI. If you use multiple AI agents, Cresta.AI uses **X-headers** to differentiate routing within its platform.

The following example shows a transfer number configuration.

The screenshot shows a window titled "iva_transfer_number Properties". It has two tabs: "General" and "Reporting". The "General" tab is selected. Inside the "General" tab, there is a text box containing "iva_transfer_number". Below it is a "Description:" label followed by a text box containing "Set the Default Value of this CAV with the number provided by the Vendor". There are two checkboxes: "Reporting Call Variable (Saved in Database)" and "Field Contains Sensitive Data", both of which are unchecked. Below these is a "Data Type:" label followed by a dropdown menu showing "String". At the bottom of the "General" tab is a "Restrictions" section with two radio buttons: "Undefined" and "Value". The "Value" radio button is selected. To the right of the "Value" radio button is a text box containing "9995550000". At the bottom right of the window are three buttons: "Apply", "Save", and "Cancel".

Header Usage

The Five9 implementation team manages **X-headers** in the IVA. The Five9 platform sends and receives **X-headers** during call processing, and the team can add more headers in either direction if needed. When adding custom headers, the team limits them to 10 to avoid packet fragmentation or call quality issues.

Headers Sent to the IVA

The Five9 platform sends the following platform call variables to the IVA:

- **call.ANI**
- **call.DNIS**
- **call.call_id**
- **call.session_id**
- **call.campaign_name**
- **call.campaign_id**

- **call.domain_id**
- **call.start_timestamp**

Note: The Five9 implementation team can edit the SetHeaders module to add more variables.

Headers Received from the IVA

The Five9 platform receives the following headers from the IVA:

- **X-RouteReason**
- **X-RouteType**
- **X-RouteValue**
- **X-CustomMetaData**

Note: The Five9 implementation team can edit the SetHeaders module to add more headers.

Decagon Implementation Guide

This section explains integrating Decagon, a certified third-party AI provider, with the Five9 platform.

Five9 offers integration services but doesn't resell the Decagon product. You must purchase and implement Decagon through Decagon.

Prerequisites

Before you begin, make sure you have:

- A Five9-provisioned domain
- A Decagon-activated account

- A Decagon-assigned phone number
- Decagon-generated query credentials
- A Decagon-provided SIP domain value (FQDN)

Integration Process

The following tasks outline what each party does during the Decagon integration.

What Decagon Does (Informational Only)

- 1 Creates your account in the Decagon platform.
- 2 Provides the integration details required to begin setup.
- 3 Uses **X-headers** to differentiate routing to its AI platform when multiple AI agents exist.

What You Do (Customer Responsibility)

- 1 Provide your Decagon phone number (one per customer) to the Five9 implementation team.
- 2 Provide the Decagon SIP domain value required for the integration.
- 3 Provide your Decagon query credentials, including the following information per customer:
 - Query URL
 - API key
 - Bearer token

What the Five9 Implementation Team Does (Informational Only)

- 1 Provides your domain ID to complete the IVA account setup.
- 2 Applies your AI phone number for outbound call routing through the Decagon integration.

- 3 Creates a campaign and assigns a test direct inward dialing (DID) number.
- 4 Configures call-attached variables (CAVs) to enable reporting. The delivery team will advise you if additional reporting variables are required.
- 5 Imports the Five9 IVR template using the default configuration. For more information about transfer settings, see [IVR Transfer Configuration](#).
- 6 Assigns the AI phone number to the third-party transfer module.
- 7 Sets the Decagon SIP domain value (FQDN) in the **X-FQDN-OB-PRD-DECAGON** header.
- 8 Tests the campaign to confirm that calls successfully reach Decagon.

IVR Transfer Configuration

The integration sends **X-headers** on outbound invites from the Five9 platform. The platform retrieves data from Decagon using the Query module.

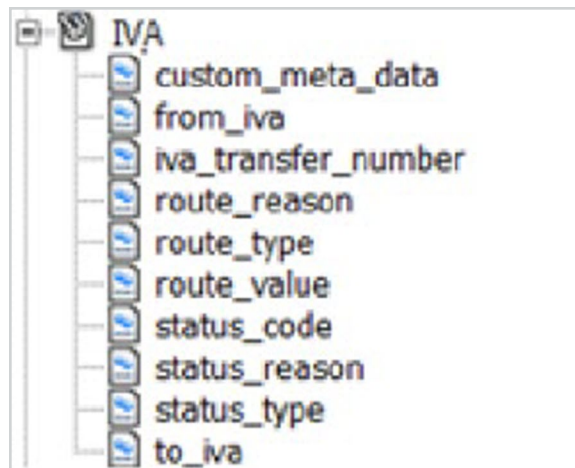
The Five9 implementation team completes the following IVR transfer configuration steps to configure the Five9 platform for call transfers that use default **X-headers**.

Note: As customer, you don't configure the IVR. The Five9 implementation team performs the IVR transfer configuration steps. You can use the following information for awareness only.

- 1 Creates a call variable group named **Decagon**.
- 2 Creates the following CAVs:

Variable	Type	Description
custom_meta_data	String	Needed return data (breadcrumb, short summary)
from_iva	String	Data returned from the IVA (KV list)
iva_transfer_number	String	AI phone number
route_reason	String	Short outcome description

Variable	Type	Description
		for Five9 reporting
route_type	String	Next action (skill transfer, phone transfer)
route_value	String	Destination (skill name, phone number)
status_type	String	Event type that triggered status (transfer, API)
status_code	Integer	Numeric error code
status_reason	String	Error reason
to_iva	String	Data sent to the IVA (KV list)

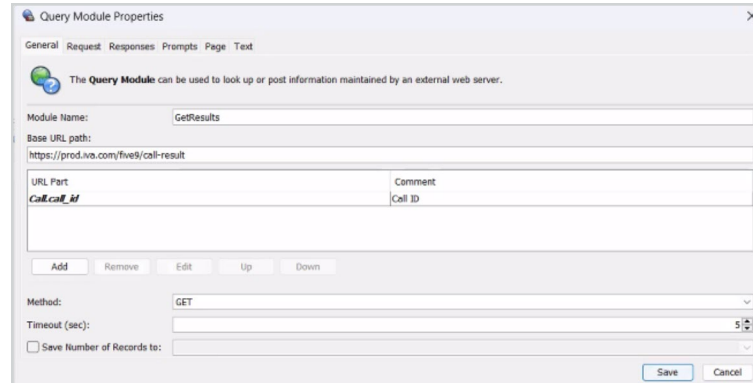


Note: The IVR script automatically populates these CAV values from corresponding **X-headers** in the API. For more information about headers, see [Header Usage](#).

- 3 Configures authentication for the Query module by modifying the request headers because Decagon doesn't support authentication profiles.
 - a Creates an authentication configuration using your Decagon query credentials.
 - b Uses the Query module instead of an Auth Profile.

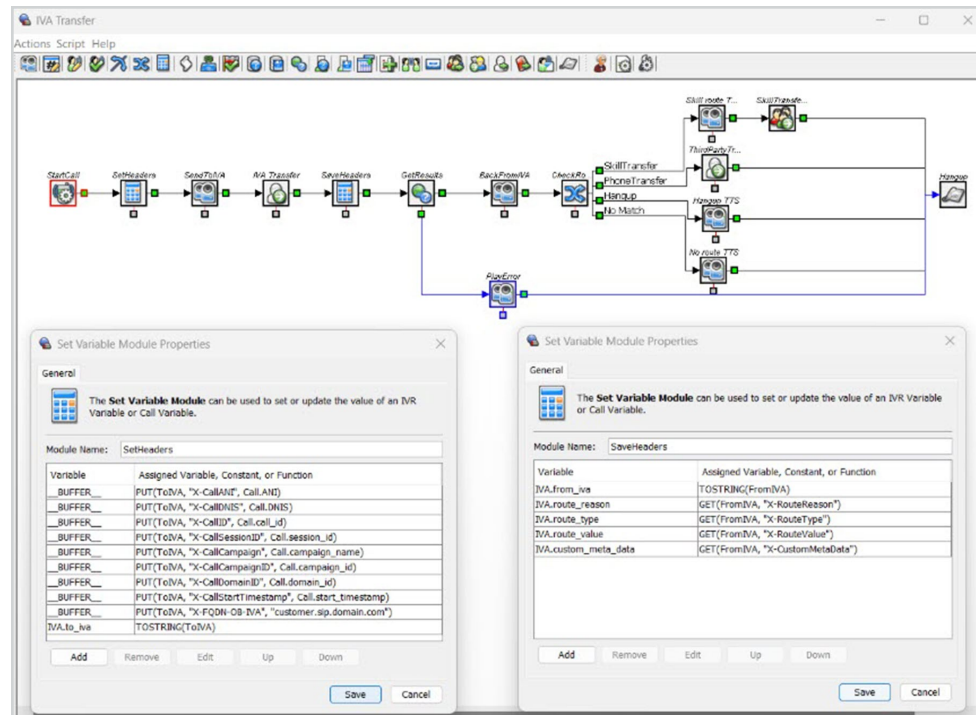
- c Modifies the Authorization header to include the Decagon-generated Bearer token.

The system already populates the production URL in the IVR template. The following example shows a pre-populated production URL in the IVR template.



4 Imports the Decagon IVR Transfer Template v1 IVR script.

The following example shows an IVR transfer script.



Note: The Five9 implementation team uses the template designed specifically for Decagon, as **X-header** names vary across integrations.

- 5 Configures the transfer number by using the single AI phone number (real or pseudo) that Decagon provides for your organization. This configuration supports one phone number per customer, regardless of the number of domains. The Five9 implementation team sets the default value of the **iva_transfer_number** variable to this phone number.

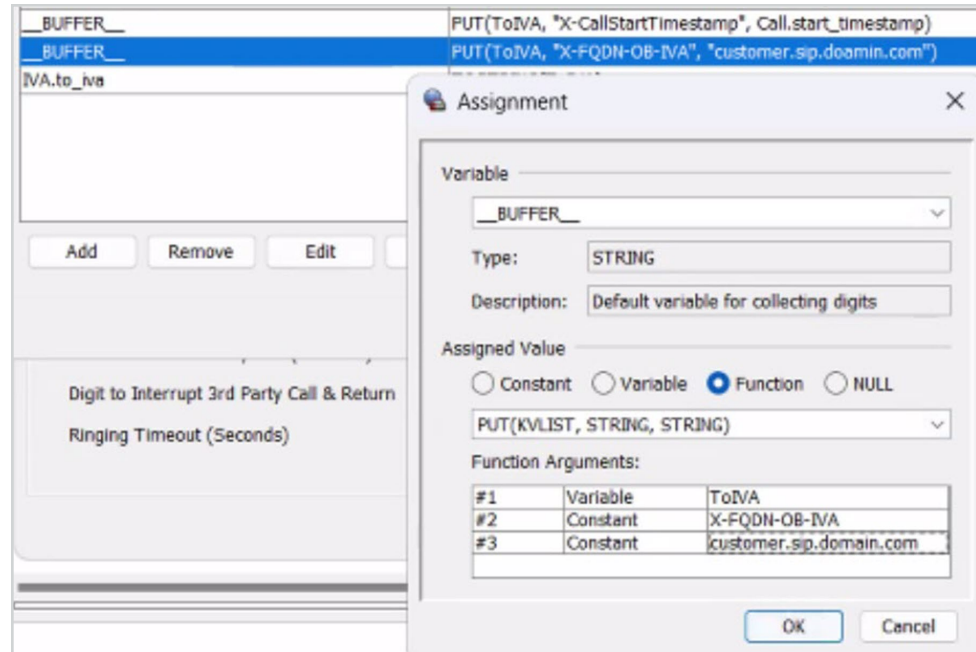
Note: The Connectivity Project Manager manages call routing for the AI phone number to the IVA integration. The Five9 implementation team also uses the same transfer DID across all your domains that send calls to Decagon. If you use multiple AI agents, Decagon uses **X-headers** to differentiate routing within its platform.

The following example shows a transfer number configuration.

The screenshot shows the 'iva_transfer_number Properties' dialog box. The 'General' tab is selected. The variable name is 'iva_transfer_number'. The description is 'Set the Default Value of this CAV with the number provided by the Vendor'. The 'Reporting Call Variable (Saved in Database)' checkbox is unchecked. The 'Field Contains Sensitive Data' checkbox is unchecked. The 'Data Type' is set to 'String'. Under the 'Restrictions' tab, the 'Value' radio button is selected, and the value '9995550000' is entered in the text field. The 'Apply', 'Save', and 'Cancel' buttons are at the bottom right.

- 6 Sets the **X-FQDN-OB-PRD-DECAGON** header to the SIP domain value provided by Decagon. The Five9 implementation team uses the unique SIP domain assigned to your organization to ensure successful call routing through the Decagon integration.

The following example shows the unique SIP domain in the Set Variable module to enable call routing.



Header Usage

The Five9 implementation team manages **X-headers** in the IVA. The Five9 platform sends and receives **X-headers** during call processing, and the team can add more headers in either direction if needed. When adding custom headers, the team limits them to 10 to avoid packet fragmentation or call quality issues.

Headers Sent to the IVA

The Five9 platform sends the following platform call variables to the IVA:

- call.ANI
- call.DNIS
- call.call_id
- call.session_id
- call.campaign_name
- call.campaign_id

- **call.domain_id**
- **call.start_timestamp**

Note: The Five9 implementation team can edit the SetHeaders module to add more variables.

Headers Received from the IVA

The Five9 platform receives the following headers from the IVA:

- **X-RouteReason**
- **X-RouteType**
- **X-RouteValue**
- **X-CustomMetaData**

Note: The Five9 implementation team can edit the SetHeaders module to add more headers.

FlipCX Implementation Guide

This section explains integrating FlipCX, a certified third-party AI provider, with the Five9 platform.

Five9 offers integration services but doesn't resell the FlipCX product. You must purchase and implement FlipCX through FlipCX.

Prerequisites

Before you begin, make sure you have:

- A Five9-provisioned domain
- A FlipCX-activated account

- A FlipCX-assigned phone number
- FlipCX-generated query credentials

Integration Process

The following tasks outline what each party does during the FlipCX integration.

What FlipCX Does (Informational Only)

- 1 Creates your account in the FlipCX platform.
- 2 Provides the integration details required to begin setup.
- 3 Uses **X-headers** to differentiate routing to its AI platform when multiple AI agents exist.

What You Do (Customer Responsibility)

- 1 Provide your FlipCX AI phone number (one per customer) to the Five9 implementation team.
- 2 Provide your FlipCX query credentials, when applicable, including the API username and password for the Query module.

What the Five9 Implementation Team Does (Informational Only)

- 1 Provides your domain ID to complete the IVA account setup.
- 2 Applies your AI phone number for outbound call routing through the FlipCX integration.
- 3 Creates a campaign and assigns a test direct inward dialing (DID) number.
- 4 Configures call-attached variables (CAVs) to enable reporting. The delivery team will advise you if additional reporting variables are required.
- 5 Imports the Five9 IVR template using the default configuration. For more information about transfer settings, see [IVR Transfer Configuration](#).

- 6 Applies your AI phone number to the third-party transfer module.
- 7 Tests the campaign to confirm that calls successfully reach FlipCX.

Note: The system returns **X-headers** when the call disconnects (BYE) and the Query module retrieves hang-up data.

IVR Transfer Configuration

The integration sends **X-headers** on outbound invites from the Five9 platform. The platform retrieves data from FlipCX using the Query module.

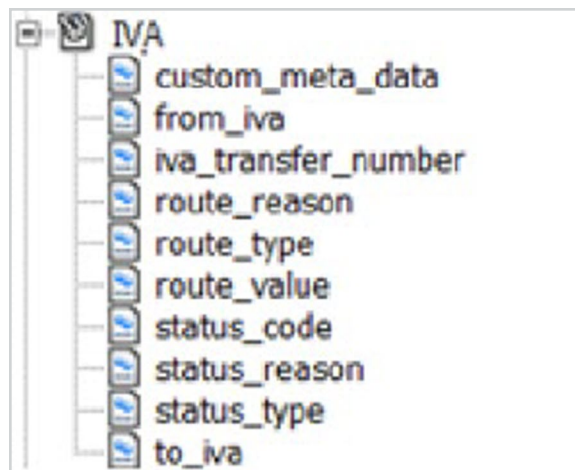
The Five9 implementation team completes the following IVR transfer configuration steps to configure the Five9 platform for call transfers that use default **X-headers**.

Note: As customer, you don't configure the IVR. The Five9 implementation team performs the IVR transfer configuration steps. You can use the following information for awareness only.

- 1 Creates a call variable group named **Flip**.
- 2 Creates the following CAVs:

Variable	Type	Description
custom_meta_data	String	Needed return data (breadcrumb, short summary)
from_iva	String	Data returned from the IVA (KV list)
iva_transfer_number	String	AI phone number
route_reason	String	Short outcome description for Five9 reporting
route_type	String	Next action (skill transfer, phone transfer)
route_value	String	Destination (skill name, phone number)

Variable	Type	Description
status_type	String	Event type that triggered status (transfer, API)
status_code	Integer	Numeric error code
status_reason	String	Error reason
to_iva	String	Data sent to the IVA (KV list)

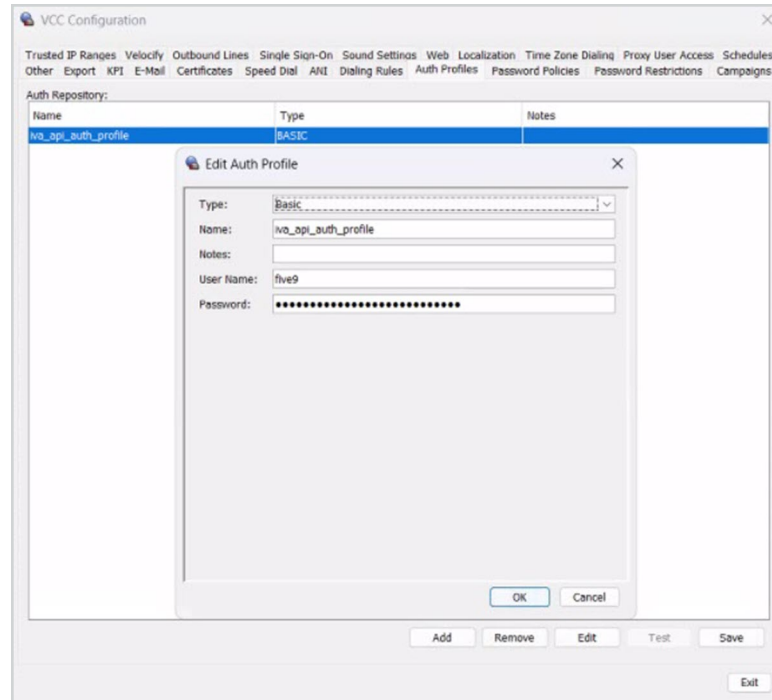


Note: The IVR script automatically populates these CAV values from corresponding **X-headers** in the API. For more information about headers, see [Header Usage](#).

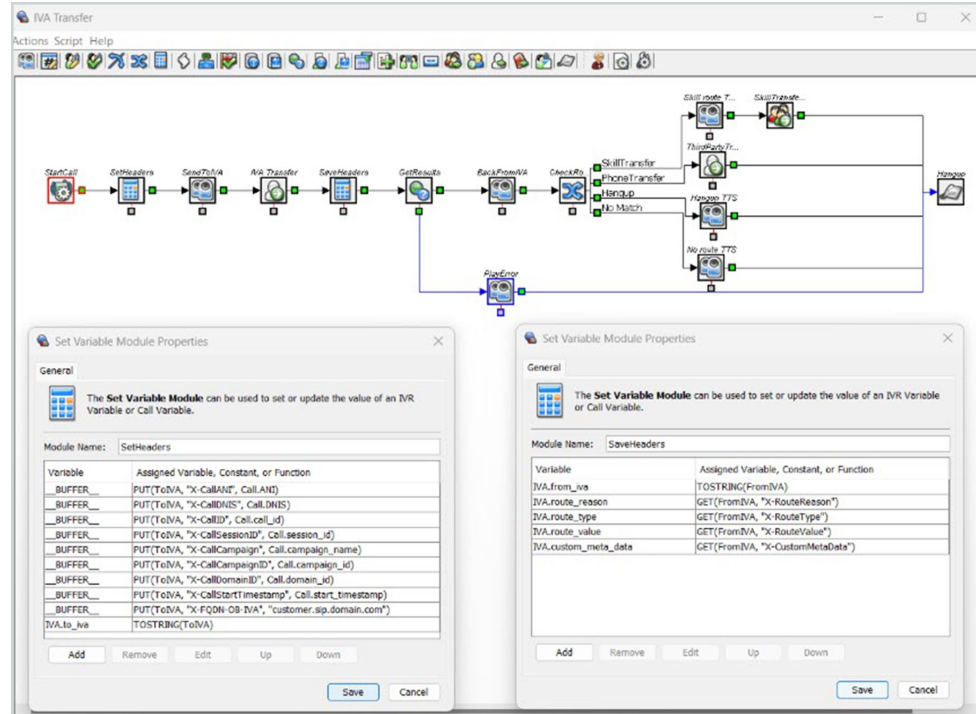
- 3** Configures the Auth Profile. If the integration requires API use, the Five9 implementation team:
 - a** Creates an authentication profile using your FlipCX credentials.
 - b** Names the profile **FlipCX**.

If the integration doesn't require API access, the team skips this step.

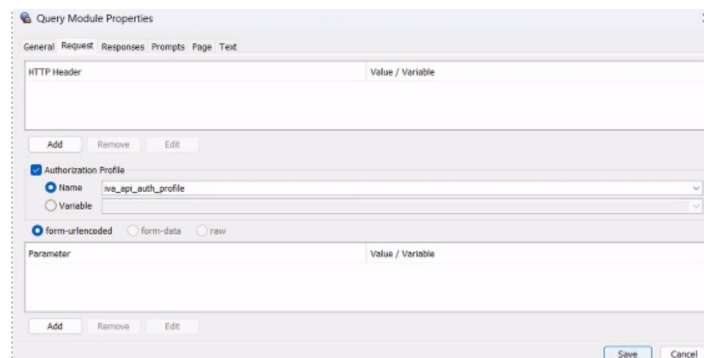
The following example shows an Auth Profile configuration.



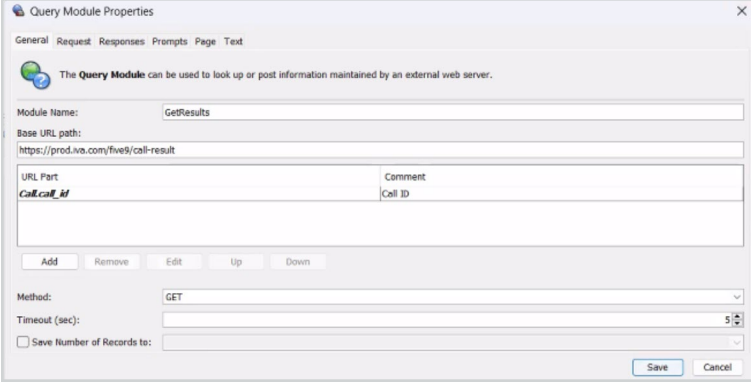
- 4 Imports the **FlipCX IVR Transfer Template v1** IVR script.
The following example shows an IVR that uses a Query module.



- 5 When applicable, assigns the FlipCX auth profile to the Query module. The following example shows an assigned auth profile to the Query module.



The system already populates the production URL in the IVR template. The following example shows a pre-populated production URL in the IVR template.



The **Query Module** can be used to look up or post information maintained by an external web server.

Module Name:

Base URL path:

URL Part	Comment
<u>Call.call_id</u>	Call ID

Method:

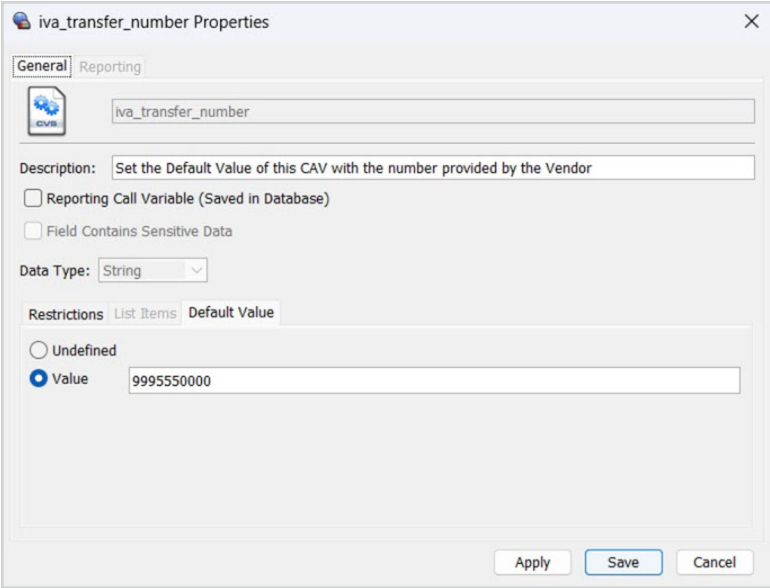
Timeout (sec):

☐ Save Number of Records to:

- 6 Configures the transfer number by using the single AI phone number that FlipCX provides for your organization. This configuration supports one phone number per customer, regardless of the number of domains. The Five9 implementation team sets the default value of the **iva_transfer_number** variable to this phone number.

Note: The Connectivity Project Manager manages call routing for the AI phone number to the IVA integration. The Five9 implementation team also uses the same transfer DID across all your domains that send calls to FlipCX. If you use multiple AI agents, FlipCX uses **X-headers** to differentiate routing within its platform.

The following example shows a transfer number configuration.



iva_transfer_number Properties

General Reporting



Description:

☐ Reporting Call Variable (Saved in Database)
☐ Field Contains Sensitive Data

Data Type:

Restrictions List Items Default Value

☐ Undefined
☒ Value

Header Usage

The Five9 implementation team manages **X-headers** in the IVA. The Five9 platform sends and receives **X-headers** during call processing, and the team can add more headers in either direction if needed. When adding custom headers, the team limits them to 10 to avoid packet fragmentation or call quality issues.

Headers Sent to the IVA

The Five9 platform sends the following platform call variables to the IVA:

- **call.ANI**
- **call.DNIS**
- **call.call_id**
- **call.session_id**
- **call.campaign_name**
- **call.campaign_id**
- **call.domain_id**
- **call.start_timestamp**

Note: The Five9 implementation team can edit the SetHeaders module to add more variables.

Headers Received from the IVA

The Five9 platform receives the following headers from the IVA:

- **X-RouteReason**
- **X-RouteType**
- **X-RouteValue**
- **X-CustomMetaData**

Note: The Five9 implementation team can edit the SetHeaders module to add more headers.

Hyro Implementation Guide

This section explains integrating Hyro, a certified third-party AI provider, with the Five9 platform.

Five9 offers integration services but doesn't resell the Hyro product. You must purchase and implement Hyro through Hyro.

Prerequisites

Before you begin, make sure you have:

- A Five9-provisioned domain
- A Hyro-activated account
- A Hyro-assigned phone number
- Hyro-generated query credentials
- A Hyro-provided SIP domain value (FQDN)

Integration Process

The following tasks outline what each party does during the Hyro integration.

What Hyro Does (Informational Only)

- 1 Creates your account in the Hyro platform.
- 2 Provides the integration details required to begin setup.

- 3 Uses **X-headers** to differentiate routing to its AI platform when multiple AI agents exist.

What You Do (Customer Responsibility)

- 1 Provide your Hyro phone number (two per customer) to the Five9 implementation team.
- 2 Provide the Hyro SIP domain values required for the integration for both Staging/Test and Production SIP domains (FQDN, customer-specific; two SIP domain values per customer).
- 3 Provide your Hyro query credentials, when applicable, including the following information per customer:
 - Static URL
 - Client ID
 - Client secret
 - Private key
 - Passphrase

What the Five9 Implementation Team Does (Informational Only)

- 1 Provides your domain ID to complete the IVA account setup.
- 2 Applies your AI phone number for outbound call routing through the Hyro integration.
- 3 Creates a campaign and assigns a test direct inward dialing (DID) number.
- 4 Configures call-attached variables (CAVs) to enable reporting. The delivery team will advise you if additional reporting variables are required.
- 5 Imports the Five9 IVR template using the default configuration. For more information about transfer settings, see [IVR Transfer Configuration](#).
- 6 Assigns the AI phone number to the third-party transfer module.

- 7 Sets the Hyro SIP domain value (FQDN) in the **X-FQDN-OB-PRD-HYRO** header.
- 8 Tests the campaign to confirm that calls successfully reach Hyro.

IVR Transfer Configuration

The integration sends **X-headers** on outbound invites from the Five9 platform. The platform retrieves data from Hyro using the Query module.

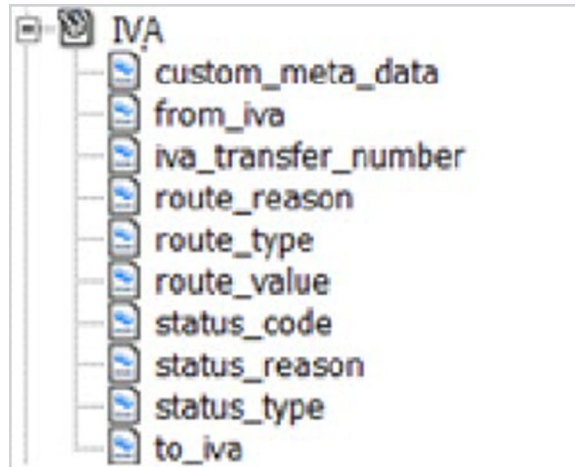
The Five9 implementation team completes the following IVR transfer configuration steps to configure the Five9 platform for call transfers that use default **X-headers**.

Note: As customer, you don't configure the IVR. The Five9 implementation team performs the IVR transfer configuration steps. You can use the following information for awareness only.

- 1 Creates a call variable group named **Hyro**.
- 2 Creates the following CAVs:

Variable	Type	Description
custom_meta_data	String	Needed return data (breadcrumb, short summary)
from_iva	String	Data returned from the IVA (KV list)
iva_transfer_number	String	AI phone number
route_reason	String	Short outcome description for Five9 reporting
route_type	String	Next action (skill transfer, phone transfer)
route_value	String	Destination (skill name, phone number)
status_type	String	Event type that triggered status (transfer, API)
status_code	Integer	Numeric error code

Variable	Type	Description
status_reason	String	Error reason
to_iva	String	Data sent to the IVA (KV list)

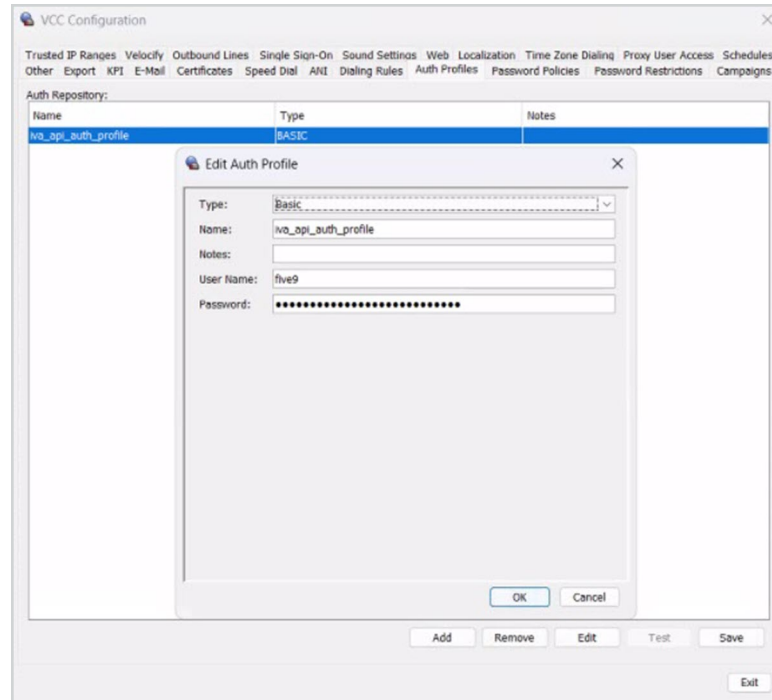


Note: The IVR script automatically populates these CAV values from corresponding **X-headers** in the API. For more information about headers, see [Header Usage](#).

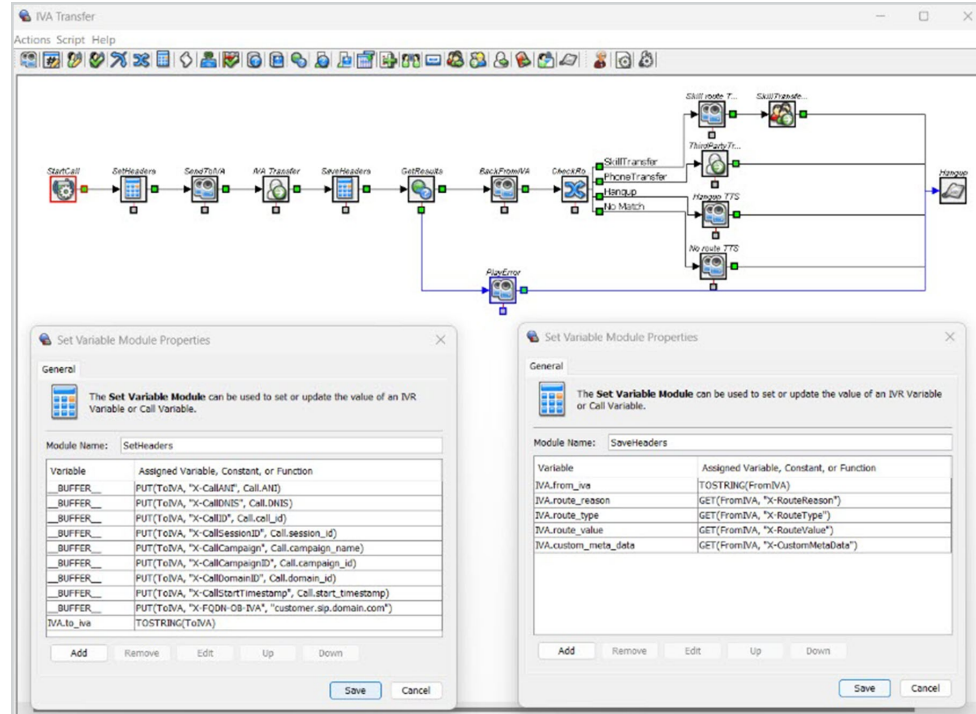
3 Configures the Auth Profile using OAuth2. To set the authentication profile, the Five9 implementation team:

- a** Creates an authentication profile using your Hyro credentials.
- b** Names the profile **Hyro**.

The following example shows an Auth Profile configuration.



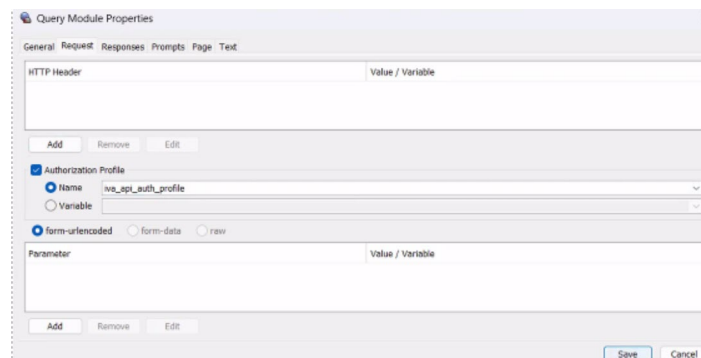
- 4 Imports the **Hyro IVR Transfer Template v1** IVR script.
The following example shows an IVR transfer script.



Note: The Five9 implementation team uses the template designed specifically for Hyro, as **X-header** names vary across integrations.

- When applicable, assigns the Hyro auth profile to the Query module. Since the Query module requires additional request headers, the Five9 implementation team also updates the request configuration by adding X-Passphrase and X-Private-Key values to the Request section of the Query module.

The following example shows an assigned auth profile to the Query module.



The system already populates the production URL in the IVR template. The following example shows a pre-populated production URL in the IVR template.

Query Module Properties

General Request Responses Prompts Page Text

The **Query Module** can be used to look up or post information maintained by an external web server.

Module Name: GetResults

Base URL path: https://prod.iva.com/five9/call-result

URL Part	Comment
CallCallId	Call ID

Add Remove Edit Up Down

Method: GET

Timeout (sec): 5

☐ Save Number of Records to:

Save Cancel

- 6 Configures the transfer number by using the single AI phone number (real or pseudo) that Hyro provides for your organization. This configuration supports one phone number per customer, regardless of the number of domains. The Five9 implementation team sets the default value of the **iva_transfer_number** variable to this phone number.

Note: The Connectivity Project Manager manages call routing for the AI phone number to the IVA integration. The Five9 implementation team also uses the same transfer DID across all your domains that send calls to Hyro. If you use multiple AI agents, Hyro uses **X-headers** to differentiate routing within its platform.

The following example shows a transfer number configuration.

iva_transfer_number Properties

General Reporting

iva_transfer_number

Description: Set the Default Value of this CAV with the number provided by the Vendor

☐ Reporting Call Variable (Saved in Database)

☐ Field Contains Sensitive Data

Data Type: String

Restrictions List Items Default Value

☐ Undefined

☒ Value 9995550000

Apply Save Cancel

- 7 Sets the **X-FQDN-OB-PRD-HYRO** header to the SIP domain value provided by Hyro. The Five9 implementation team uses the unique SIP domain assigned to your organization to ensure successful call routing through the Hyro integration. The following example shows the unique SIP domain in the Set Variable module to enable call routing.

Assignment

Variable: _BUFFER_

Type: STRING

Description: Default variable for collecting digits

Assigned Value: ☒ Constant ☐ Variable ☒ Function ☐ NULL

Function: PUT(KVLIST, STRING, STRING)

Function Arguments:

#	Type	Value
#1	Variable	ToIVA
#2	Constant	X-FQDN-OB-IVA
#3	Constant	customer.sip.domain.com

OK Cancel

Header Usage

The Five9 implementation team manages **X-headers** in the IVA. The Five9 platform sends and receives **X-headers** during call processing, and the team can add more headers in either direction if needed. When adding custom headers, the team limits them to 10 to avoid packet fragmentation or call quality issues.

Headers Sent to the IVA

The Five9 platform sends the following platform call variables to the IVA:

- **call.ANI**
- **call.DNIS**
- **call.call_id**
- **call.session_id**
- **call.campaign_name**
- **call.campaign_id**
- **call.domain_id**
- **call.start_timestamp**

Note: The Five9 implementation team can edit the SetHeaders module to add more variables.

Headers Received from the IVA

The Five9 platform receives the following headers from the IVA:

- **X-RouteReason**
- **X-RouteType**
- **X-RouteValue**
- **X-CustomMetaData**

Note: The Five9 implementation team can edit the SetHeaders module to add more headers.

Kastle.AI Implementation Guide

This section explains integrating Kastle.AI, a certified third-party AI provider, with the Five9 platform.

Five9 offers integration services but doesn't resell the Kastle.AI product. You must purchase and implement Kastle.AI through Kastle.AI.

Prerequisites

Before you begin, make sure you have:

- A Five9-provisioned domain
- A Kastle.AI-activated account
- A Kastle.AI-assigned phone number
- Kastle.AI-generated query credentials
- A Kastle.AI-provided SIP domain value (FQDN)

Integration Process

The following tasks outline what each party does during the Kastle.AI integration.

What Kastle.AI Does (Informational Only)

- 1 Creates your account in the Kastle.AI platform.
- 2 Provides the integration details required to begin setup.

- 3 Uses **X-headers** to differentiate routing to its AI platform when multiple AI agents exist.

What You Do (Customer Responsibility)

- 1 Provide your Kastle.AI phone number (one per customer) to the Five9 implementation team.
- 2 Provide the Kastle.AI SIP domain value required for the integration.
- 3 Provide your Kastle.AI query credentials, when applicable, including the API username and password for the Query module.

What the Five9 Implementation Team Does (Informational Only)

- 1 Provides your domain ID to complete the IVA account setup.
- 2 Applies your AI phone number for outbound call routing through the Kastle.AI integration.
- 3 Creates a campaign and assigns a test direct inward dialing (DID) number.
- 4 Configures call-attached variables (CAVs) to enable reporting. The delivery team will advise you if additional reporting variables are required.
- 5 Imports the Five9 IVR template using the default configuration. For more information about transfer settings, see [IVR Transfer Configuration](#).
- 6 Assigns the AI phone number to the **KastleNumber** variable.
- 7 Sets the Kastle.AI SIP domain value (FQDN) in the **X-OB-FQDN-PRD-KASTLE** header.
- 8 Tests the campaign to confirm that calls successfully reach Kastle.AI.

IVR Transfer Configuration

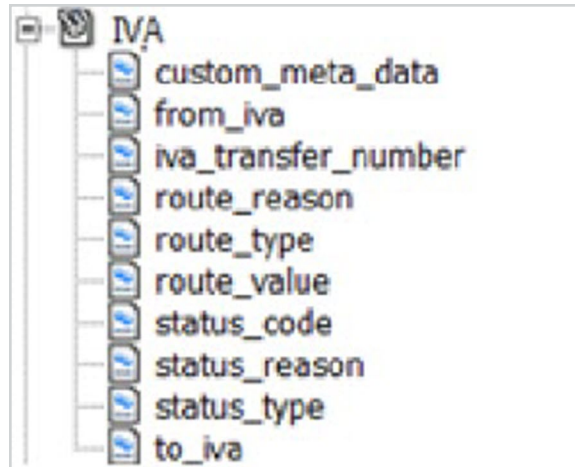
The integration sends **X-headers** on outbound invites from the Five9 platform. The platform retrieves data from Kastle.AI using the Query module.

The Five9 implementation team completes the following IVR transfer configuration steps to configure the Five9 platform for call transfers that use default **X-headers**.

Note: As customer, you don't configure the IVR. The Five9 implementation team performs the IVR transfer configuration steps. You can use the following information for awareness only.

- 1 Creates a call variable group named **KastleAI**.
- 2 Creates the following CAVs:

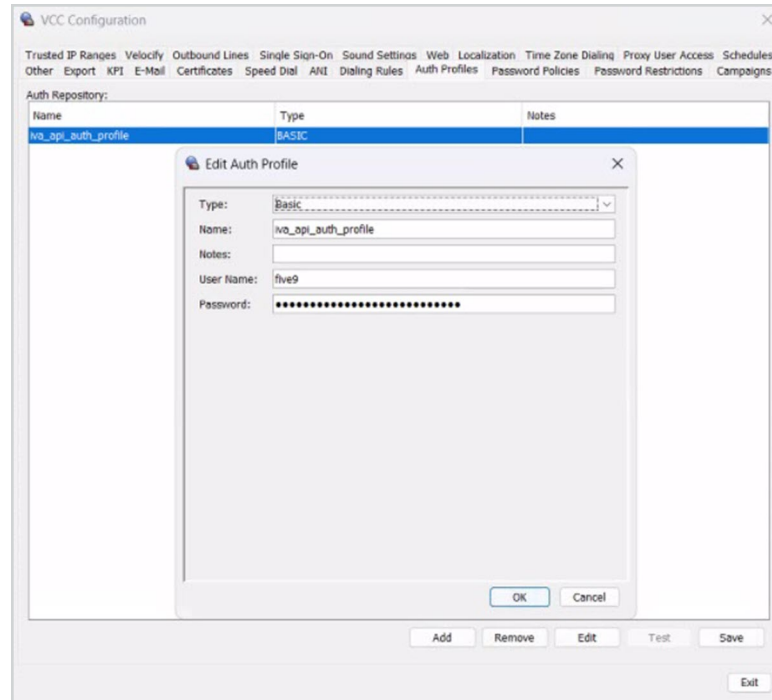
Variable	Type	Description
custom_meta_data	String	Needed return data (breadcrumb, short summary)
from_iva	String	Data returned from the IVA (KV list)
iva_transfer_number	String	AI phone number
route_reason	String	Short outcome description for Five9 reporting
route_type	String	Next action (skill transfer, phone transfer)
route_value	String	Destination (skill name, phone number)
status_type	String	Event type that triggered status (transfer, API)
status_code	Integer	Numeric error code
status_reason	String	Error reason
to_iva	String	Data sent to the IVA (KV list)



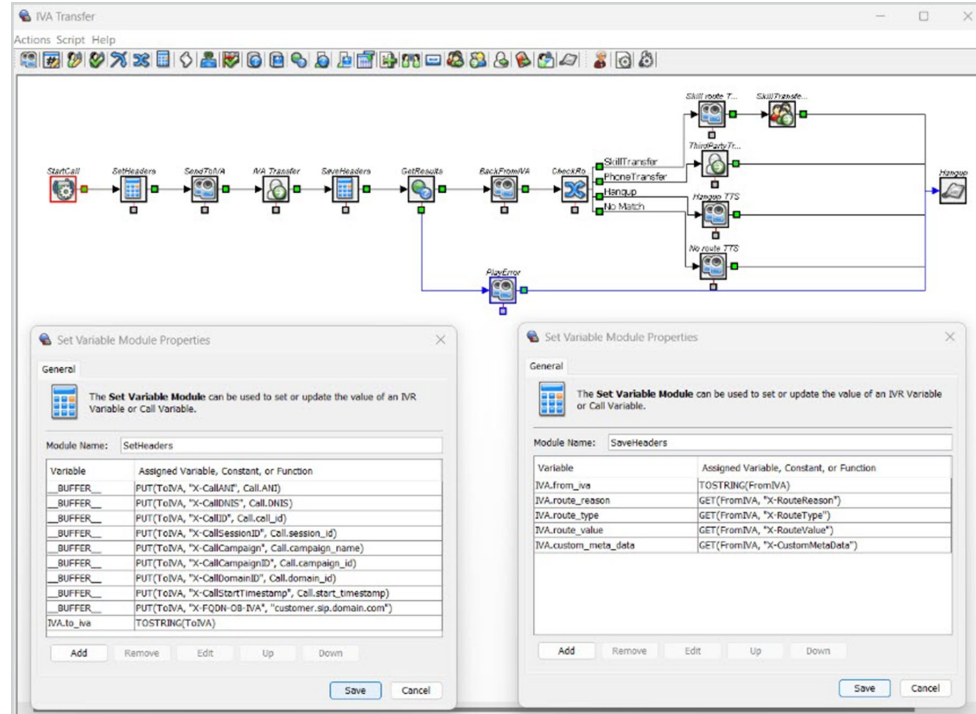
Note: The IVR script automatically populates these CAV values from corresponding **X-headers** in the API. For more information about headers, see [Header Usage](#).

- 3** Configures the Auth Profile. To set the authentication profile, the Five9 implementation team:
 - a** Creates an authentication profile using your Kastle.AI credentials.
 - b** Names the profile **Kastle.ai**.

The following example shows an Auth Profile configuration.

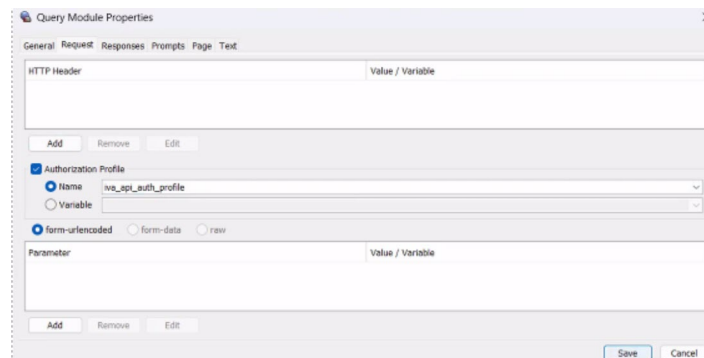


- 4 Imports the **Kastle AI Agent Connect Integration v1** IVR script.
The following example shows an IVR transfer script.

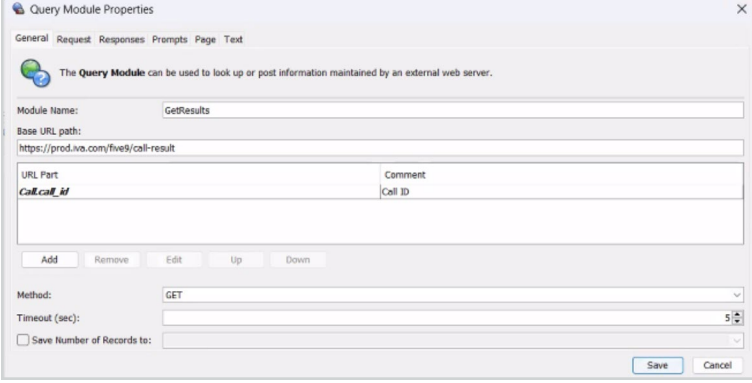


Note: The Five9 implementation team uses the template designed specifically for Kastle.AI, as **X-header** names vary across integrations.

- When applicable, assigns the Kastle.AI auth profile to the Query module. The following example shows an assigned auth profile to the Query module.



The system already populates the production URL in the IVR template. The following example shows a pre-populated production URL in the IVR template.



The **Query Module** can be used to look up or post information maintained by an external web server.

Module Name:

Base URL path:

URL Part	Comment
CallCallId	Call ID

Method:

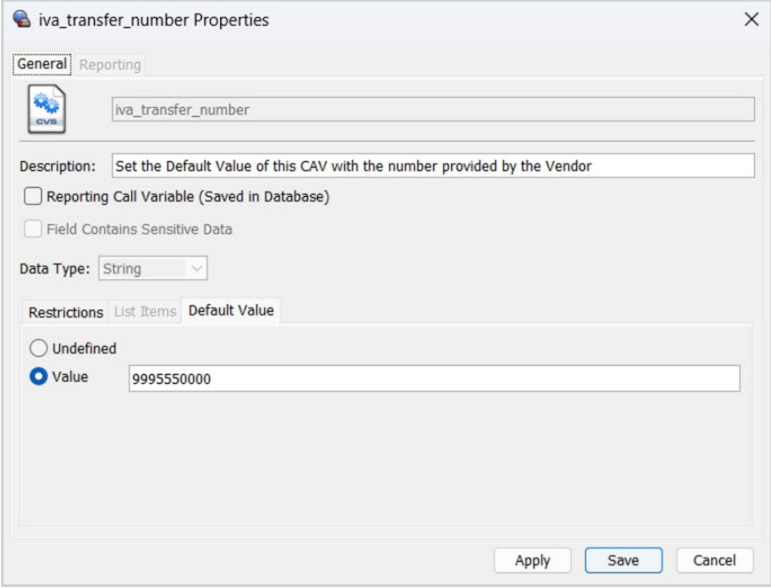
Timeout (sec):

☐ Save Number of Records to:

- 6 Configures the transfer number by using the single AI phone number that Kastle.AI provides for your organization. This configuration supports one phone number per customer, regardless of the number of domains. The Five9 implementation team sets the default value of the **iva_transfer_number** variable to this phone number.

Note: The Connectivity Project Manager manages call routing for the AI phone number to the IVA integration. The Five9 implementation team also uses the same transfer DID across all your domains that send calls to Kastle.AI. If you use multiple AI agents, Kastle.AI uses **X-headers** to differentiate routing within its platform.

The following example shows a transfer number configuration.



iva_transfer_number Properties

General Reporting

Description:

☐ Reporting Call Variable (Saved in Database)
☐ Field Contains Sensitive Data

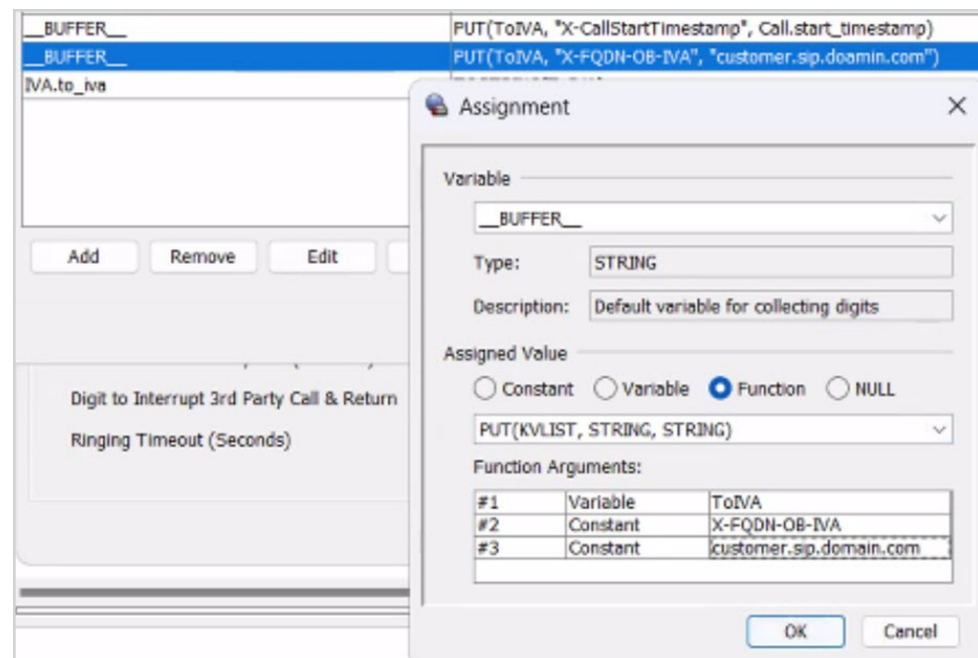
Data Type:

Restrictions List Items Default Value

☐ Undefined
☒ Value

- 7 Sets the **X-OB-FQDN-PRD-KASTLE** header to the SIP domain value provided by Kastle.AI. The Five9 implementation team uses the unique SIP domain assigned to your organization to ensure successful call routing through the Kastle.AI integration.

The following example shows the unique SIP domain in the Set Variable module to enable call routing.



Header Usage

The Five9 implementation team manages **X-headers** in the IVA. The Five9 platform sends and receives **X-headers** during call processing, and the team can add more headers in either direction if needed. When adding custom headers, the team limits them to 10 to avoid packet fragmentation or call quality issues.

Headers Sent to the IVA

The Five9 platform sends the following platform call variables to the IVA:

- call.ANI
- call.DNIS
- call.call_id

- `call.session_id`
- `call.campaign_name`
- `call.campaign_id`
- `call.domain_id`
- `call.start_timestamp`

Note: The Five9 implementation team can edit the SetHeaders module to add more variables.

Headers Received from the IVA

The Five9 platform receives the following headers from the IVA:

- `X-RouteReason`
- `X-RouteType`
- `X-RouteValue`
- `X-CustomMetaData`

Note: The Five9 implementation team can edit the SetHeaders module to add more headers.

LevelAI Implementation Guide

This section explains integrating LevelAI, a certified third-party AI provider, with the Five9 platform.

Prerequisites

Before you begin, make sure you have:

- A Five9-provisioned domain
- A Five9 domain ID
- A LevelAI-activated account
- A LevelAI direct inward dialing (DID) phone number
- A customer-specific Level-VA value

Integration Process

The following tasks outline what each party does during the LevelAI integration.

What LevelAI Does (Informational Only)

- 1 Creates your account in the LevelAI platform.
- 2 Configures Five9-provided domain ID.
- 3 Provides the integration details required to begin setup.
- 4 Uses **X-headers** to differentiate routing to its AI platform when multiple AI agents exist.

What You Do (Customer Responsibility)

- 1 Provide your LevelAI phone number (one per customer) to the Five9 implementation team.
- 2 Provide your customer-unique identifier for the call-attached variable (CAV) Level-VA.

What the Five9 Implementation Team Does (Informational Only)

- 1 Provides your domain ID to complete the IVA account setup.
- 2 Applies your AI phone number for outbound call routing through the LevelAI integration.
- 3 Creates a campaign and assigns a test DID number.

- 4 Configures CAVs to enable reporting, including the LevelAI-provided Level-VA CAV values.

The delivery team will advise you if additional reporting variables are required.

- 5 Imports the Five9 IVR template using the default configuration. For more information about transfer settings, see [IVR Transfer Configuration](#).
- 6 Applies your DID phone number to the third-party transfer module.
- 7 Tests the campaign to confirm that calls successfully reach LevelAI.

IVR Transfer Configuration

The integration sends **X-headers** on outbound invites from the Five9 platform. The platform retrieves data from LevelAI using the Query module.

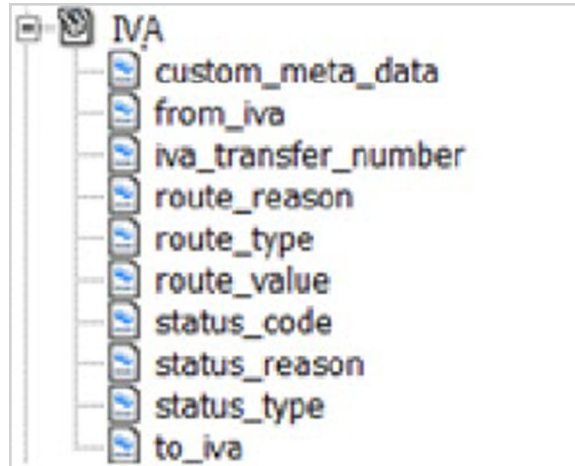
The Five9 implementation team completes the following IVR transfer configuration steps to configure the Five9 platform for call transfers that use default **X-headers**.

Note: As customer, you don't configure the IVR. The Five9 implementation team performs the IVR transfer configuration steps. You can use the following information for awareness only.

- 1 Creates a call variable group named **LevelAI**.
- 2 Creates the following CAVs:

Variable	Type	Description
custom_meta_data	String	Needed return data (breadcrumb, short summary)
from_iva	String	Data returned from the IVA (KV list)
iva_transfer_number	String	AI phone number
route_reason	String	Short outcome description for Five9 reporting
route_type	String	Next action (skill transfer, phone transfer)

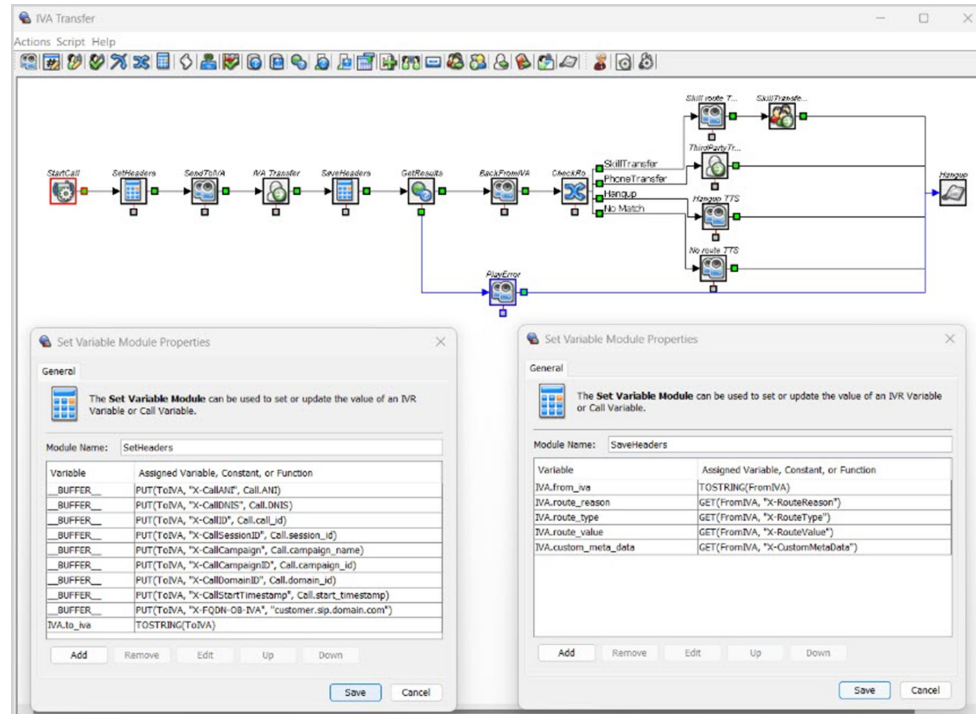
Variable	Type	Description
route_value	String	Destination (skill name, phone number)
status_type	String	Event type that triggered status (transfer, API)
status_code	Integer	Numeric error code
status_reason	String	Error reason
to_iva	String	Data sent to the IVA (KV list)
Level-VA	String	Customer-specific



Note: The IVR script automatically populates these CAV values from corresponding **X-headers** in the API. For more information about headers, see [Header Usage](#).

3 Imports the **LevelAI IVR Transfer Template v1** IVR script.

The following example shows an IVR transfer script.



Note: The Five9 implementation team uses the template designed specifically for LevelAI, as **X-header** names vary across integrations.

- 4 Configures the transfer number by using the single AI phone number (real or pseudo) that LevelAI provides for your organization. This configuration supports one phone number per customer, regardless of the number of domains. The Five9 implementation team sets the default value of the **iva_transfer_number** variable to this phone number.

Note: The Connectivity Project Manager manages call routing for the AI phone number to the IVA integration. The Five9 implementation team also uses the same transfer DID across all your domains that send calls to LevelAI. If you use multiple AI agents, LevelAI uses **X-headers** to differentiate routing within its platform.

The following example shows a transfer number configuration.

The screenshot shows a dialog box titled "iva_transfer_number Properties". It has two tabs: "General" and "Reporting". The "General" tab is selected. Inside, there is a text field for the variable name "iva_transfer_number". Below it is a description: "Set the Default Value of this CAV with the number provided by the Vendor". There are two checkboxes: "Reporting Call Variable (Saved in Database)" and "Field Contains Sensitive Data", both of which are unchecked. A "Data Type" dropdown menu is set to "String". Below this, there are three tabs: "Restrictions", "List Items", and "Default Value". The "Restrictions" tab is selected, showing two radio buttons: "Undefined" and "Value". The "Value" radio button is selected, and next to it is a text field containing the number "9995550000". At the bottom right of the dialog are three buttons: "Apply", "Save", and "Cancel".

Header Usage

The Five9 implementation team manages **X-headers** in the IVA. The Five9 platform sends and receives **X-headers** during call processing, and the team can add more headers in either direction if needed. When adding custom headers, the team limits them to 10 to avoid packet fragmentation or call quality issues.

Headers Sent to the IVA

The Five9 platform sends the following platform call variables to the IVA:

- **call.ANI**
- **call.DNIS**
- **call.call_id**
- **call.session_id**
- **call.campaign_name**
- **call.campaign_id**
- **call.domain_id**

- **call.start_timestamp**
- **X-Level-VA**

Note: The Five9 implementation team can edit the SetHeaders module to add more variables.

Headers Received from the IVA

The Five9 platform receives the following headers from the IVA:

- **X-RouteReason**
- **X-RouteType**
- **X-RouteValue**
- **summary**

Note: The Five9 implementation team can edit the SetHeaders module to add more headers.

Parloa.AI Implementation Guide

This section explains integrating Parloa.AI, a certified third-party AI provider, with the Five9 platform.

Five9 offers integration services but doesn't resell the Parloa.AI product. You must purchase and implement Parloa.AI through Parloa.AI.

Prerequisites

Before you begin, make sure you have:

- A Five9-provisioned domain
- A Parloa.AI-activated account

- A Parloa.AI-assigned phone number
- Parloa.AI-generated query credentials
- A Parloa.AI-provided SIP domain value (FQDN)

Integration Process

The following tasks outline what each party does during the Parloa.AI integration.

What Parloa.AI Does (Informational Only)

- 1 Creates your account in the Parloa.AI platform.
- 2 Provides the integration details required to begin setup.
- 3 Provides a block of five pseudo numbers for each customer domain to reduce the number of routing changes.
- 4 Uses **X-headers** to differentiate routing to its AI platform when multiple AI agents exist.

What You Do (Customer Responsibility)

- 1 Provide your Parloa.AI phone number (five per customer) to the Five9 implementation team.
- 2 Provide the Parloa.AI SIP domain value required for the integration.
- 3 Provide your Parloa.AI query credentials, when applicable, including the API username and password for the Query module.

What the Five9 Implementation Team Does (Informational Only)

- 1 Applies your AI phone number for outbound call routing through the Parloa.AI integration.
- 2 Creates a campaign and assigns a test direct inward dialing (DID) number.
- 3 Configures call-attached variables (CAVs) to enable reporting. The delivery team will advise you if additional reporting variables are required.

- 4 Imports the Five9 IVR template using the default configuration. For more information about transfer settings, see [IVR Transfer Configuration](#).
- 5 Assigns the AI phone number to the third-party transfer module.
- 6 Sets the Parloa.AI SIP domain value (FQDN) in the **X-FQDN-OB-PARLOA** header.
- 7 Tests the campaign to confirm that calls successfully reach Parloa.AI.

IVR Transfer Configuration

The integration sends **X-headers** on outbound invites from the Five9 platform. The platform retrieves data from Parloa.AI using the Query module.

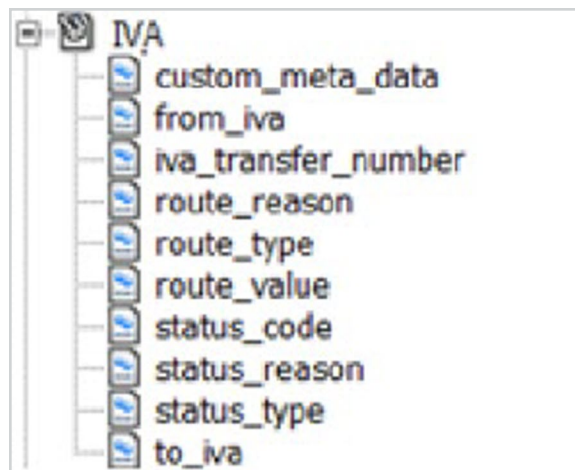
The Five9 implementation team completes the following IVR transfer configuration steps to configure the Five9 platform for call transfers that use default **X-headers**.

Note: As customer, you don't configure the IVR. The Five9 implementation team performs the IVR transfer configuration steps. You can use the following information for awareness only.

- 1 Creates a call variable group named **Parloa**.
- 2 Creates the following CAVs:

Variable	Type	Description
custom_meta_data	String	Needed return data (breadcrumb, short summary)
from_iva	String	Data returned from the IVA (KV list)
iva_transfer_number	String	AI phone number
route_reason	String	Short outcome description for Five9 reporting
route_type	String	Next action (skill transfer, phone transfer)
route_value	String	Destination (skill name, phone number)

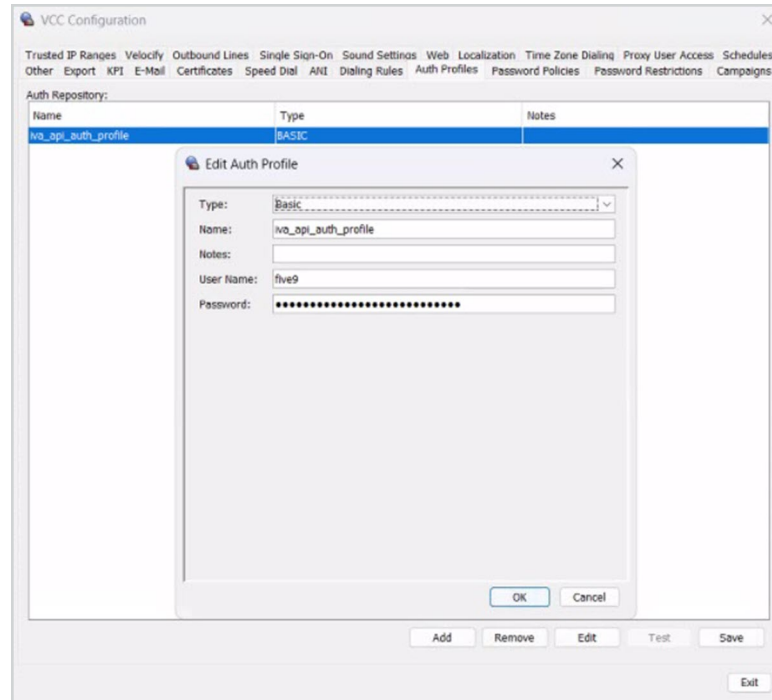
Variable	Type	Description
status_type	String	Event type that triggered status (transfer, API)
status_code	Integer	Numeric error code
status_reason	String	Error reason
to_iva	String	Data sent to the IVA (KV list)



Note: The IVR script automatically populates these CAV values from corresponding **X-headers** in the API. For more information about headers, see [Header Usage](#).

- 3** Configures the Auth Profile. To set the authentication profile, the Five9 implementation team:
 - a** Creates an authentication profile using your Parloa.AI credentials.
 - b** Names the profile **Parloa.AI**.

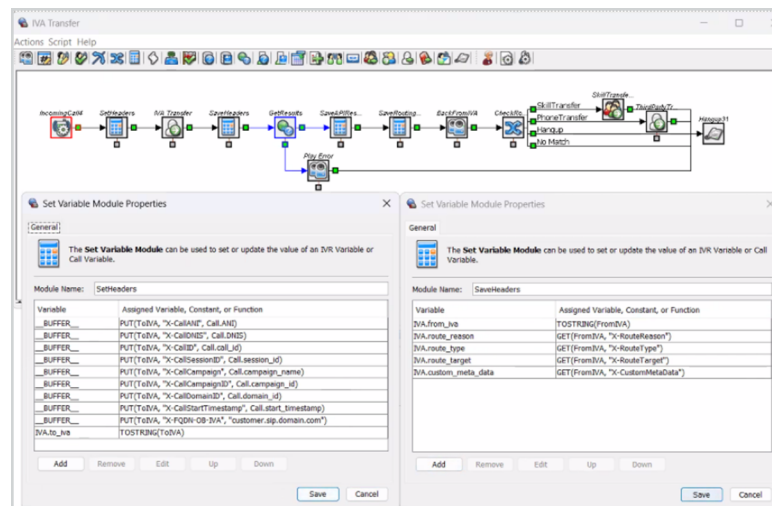
The following example shows an Auth Profile configuration.



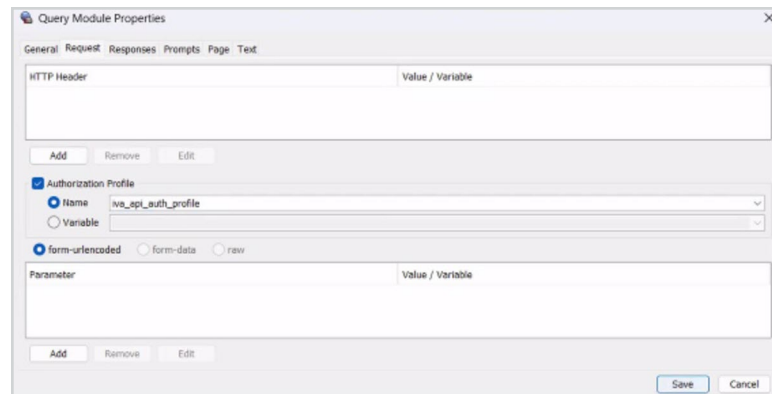
4 Imports the Parloa.AI IVR Transfer Template v1 IVR script.

Note: The Five9 implementation team uses the template designed specifically for Parloa.AI, as **X-header** names vary across integrations.

The following example shows an IVR transfer script.

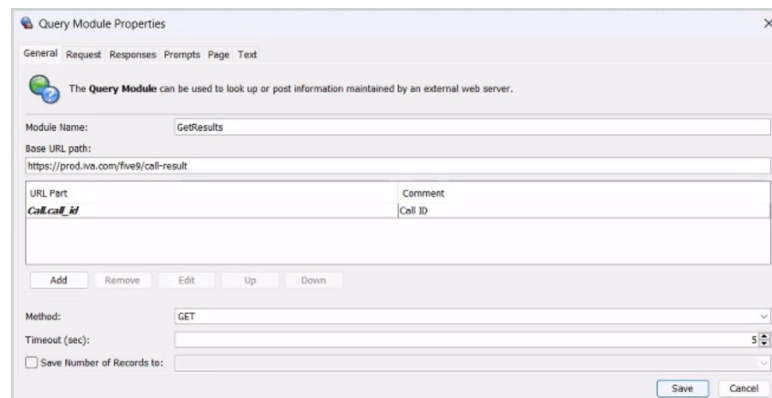


- 5 When applicable, assigns the Parloa.AI auth profile to the Query module. The following example shows an assigned auth profile to the Query module.



The screenshot shows the 'Query Module Properties' dialog box with the 'General' tab selected. The 'Authorization Profile' section is checked, and the 'Name' is set to 'iva_api_auth_profile'. The 'form-urlencoded' radio button is selected. The 'Parameter' section is empty.

The system already populates the production URL in the IVR template. The following example shows a pre-populated production URL in the IVR template.



The screenshot shows the 'Query Module Properties' dialog box with the 'General' tab selected. The 'Module Name' is 'GetResults', the 'Base URL path' is 'https://prod.iva.com/five9/call-result', and the 'Method' is 'GET'. The 'URL Part' section shows 'Call ID' as the URL part and 'Call ID' as the comment.

- 6 Configures the transfer numbers that Parloa.AI provides for your organization. The Five9 implementation team sets the default value of the **iva_transfer_number** variable to this phone number.

Important: With this configuration, call routing from Five9 requires an update each time Parloa creates a new phone number for your organization in its platform.

Note: The Connectivity Project Manager manages call routing for the AI phone numbers to the IVA integration. The Five9 implementation team also uses the same transfer DID across all your domains that send calls

to Parloa.AI. If you use multiple AI agents, Parloa.AI uses **X-headers** to differentiate routing within its platform.

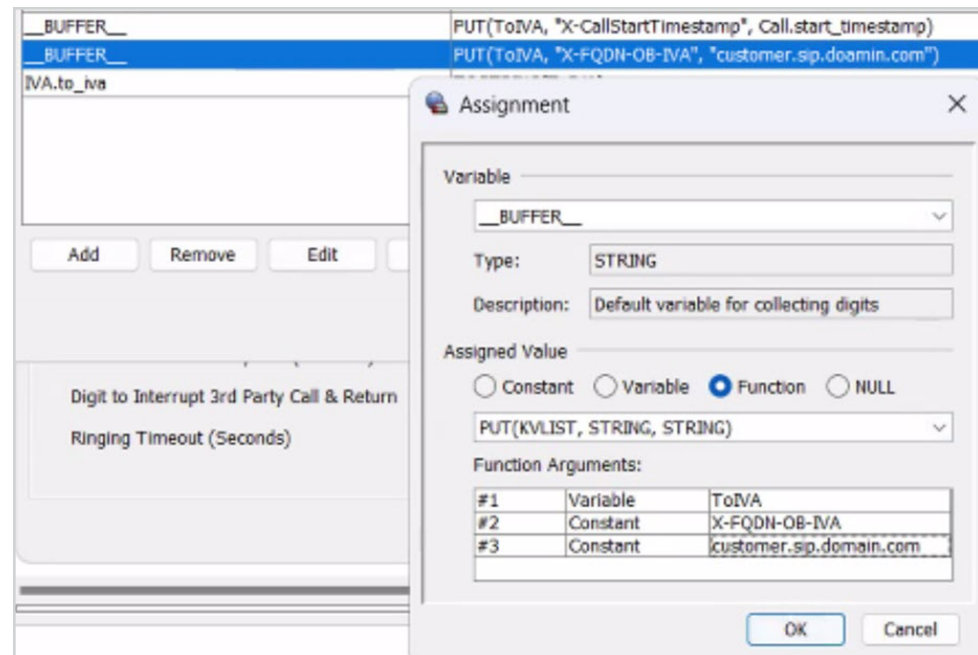
The following example shows a transfer number configuration.

The screenshot shows a dialog box titled "iva_transfer_number Properties" with a close button (X) in the top right corner. The "General" tab is selected, and the "Reporting" tab is also visible. The dialog contains the following fields and options:

- A text field labeled "iva_transfer_number" with a file icon to its left.
- A "Description:" field with the text "Set the Default Value of this CAV with the number provided by the Vendor".
- Two checkboxes: "Reporting Call Variable (Saved in Database)" and "Field Contains Sensitive Data", both of which are unchecked.
- A "Data Type:" dropdown menu set to "String".
- A section titled "Restrictions" with sub-tabs "List Items" and "Default Value". The "Default Value" sub-tab is active, showing two radio buttons: "Undefined" (unchecked) and "Value" (checked).
- A text field next to the "Value" radio button containing the number "9995550000".
- At the bottom right, there are three buttons: "Apply", "Save", and "Cancel".

- 7 Sets the **X-FQDN-OB-PARLOA** header to the SIP domain value provided by Parloa.AI. The Five9 implementation team uses the unique SIP domain assigned to your organization to ensure successful call routing through the Parloa.AI integration.

The following example shows the unique SIP domain in the Set Variable module to enable call routing.



Header Usage

The Five9 implementation team manages **X-headers** in the IVA. The Five9 platform sends and receives **X-headers** during call processing, and the team can add more headers in either direction if needed. When adding custom headers, the team limits them to 10 to avoid packet fragmentation or call quality issues.

Headers Sent to the IVA

The Five9 platform sends the following platform call variables to the IVA:

- call.ANI
- call.DNIS
- call.call_id
- call.session_id
- call.campaign_name
- call.campaign_id

- **call.domain_id**
- **call.start_timestamp**

Note: The Five9 implementation team can edit the SetHeaders module to add more variables.

Headers Received from the IVA

The Five9 platform receives the following headers from the IVA:

- **X-RouteReason**
- **X-RouteType**
- **X-RouteValue**
- **X-CustomMetaData**

Note: The Five9 implementation team can edit the SetHeaders module to add more headers.

Sierra Implementation Guide

This section explains integrating Sierra, a certified third-party AI provider, with the Five9 platform.

Five9 offers integration services but doesn't resell the Sierra product. You must purchase and implement Sierra through Sierra.

Prerequisites

Before you begin, make sure you have:

- A Five9-provisioned domain
- A Sierra-activated account

- A Sierra-assigned phone number
- Sierra-generated query credentials
- A Sierra-provided SIP domain value (FQDN)

Integration Process

The following tasks outline what each party does during the Sierra integration.

What Sierra Does (Informational Only)

- 1 Creates your account in the Sierra platform.
- 2 Provides the integration details required to begin setup.
- 3 Uses **X-headers** to differentiate routing to its AI platform when multiple AI agents exist.

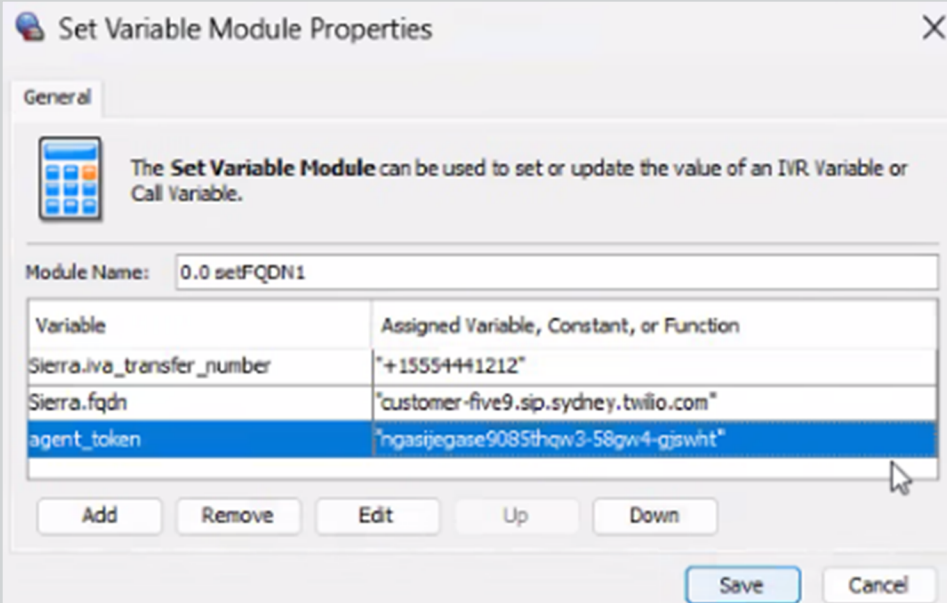
What You Do (Customer Responsibility)

- 1 Provide your Sierra phone number (multiple per customer) to the Five9 implementation team.
- 2 Provide the Sierra SIP domain value required for the integration.
- 3 Provide your Sierra query credentials, when applicable, including the following information per customer:
 - Static URL
 - Authentication token (per AI agent)
 - Agent token (per AI agent)

What the Five9 Implementation Team Does (Informational Only)

- 1 Provides your domain ID to complete the IVA account setup.
- 2 Applies your AI phone number for outbound call routing through the Sierra integration.

- 3 Creates a campaign and assigns a test direct inward dialing (DID) number.
- 4 Configures call-attached variables (CAVs) to enable reporting. The delivery team will advise you if additional reporting variables are required.
- 5 Imports the Five9 IVR template using the default configuration. For more information about transfer settings, see [IVR Transfer Configuration](#).
- 6 Assigns the AI phone number to the third-party transfer module.
- 7 Sets the Sierra SIP domain value (FQDN) in the **X-FQDN-OB-PRD-SIERRA** header.
- 8 Applies your AI phone number and the Sierra SIP domain value (FQDN) to the Set Variable Module named **0.0**.



Set Variable Module Properties

General

The **Set Variable Module** can be used to set or update the value of an IVR Variable or Call Variable.

Module Name: 0.0 setFQDN1

Variable	Assigned Variable, Constant, or Function
Sierra.iva_transfer_number	"*+15554441212"
Sierra.fqdn	"customer-five9.sip.sydney.twilio.com"
agent_token	"ngasijegase9085thow3-58gw4-gjswht"

Buttons: Add, Remove, Edit, Up, Down, Save, Cancel

- 9 Tests the campaign to confirm that calls successfully reach Sierra.

Note: After testing completes, the implementation team notifies the Connectivity Project Manager (PM) of the call outcome. If a call fails, the implementation team verifies the transfer phone number, FQDN value (don't include the @ character), and Sierra AI agent readiness. If these checks don't resolve the issue, the implementation team contacts the Connectivity PM to confirm that number routing is in place and that the

configured phone numbers are correct. If routing is correct, the assigned Connectivity PM assists with further troubleshooting.

IVR Transfer Configuration

The integration sends **X-headers** on outbound invites from the Five9 platform. The platform retrieves data from Sierra using the Query module.

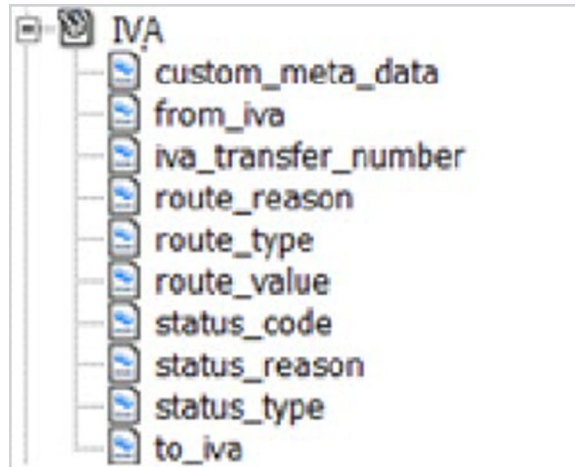
The Five9 implementation team completes the following IVR transfer configuration steps to configure the Five9 platform for call transfers that use default **X-headers**.

Note: As customer, you don't configure the IVR. The Five9 implementation team performs the IVR transfer configuration steps. You can use the following information for awareness only.

- 1 Creates a call variable group named **Sierra**.
- 2 Creates the following CAVs:

Variable	Type	Description
custom_meta_data	String	Needed return data (breadcrumb, short summary)
from_iva	String	Data returned from the IVA (KV list)
iva_transfer_number	String	AI phone number
route_reason	String	Short outcome description for Five9 reporting
route_type	String	Next action (skill transfer, phone transfer)
route_value	String	Destination (skill name, phone number)
status_type	String	Event type that triggered status (transfer, API)
status_code	Integer	Numeric error code

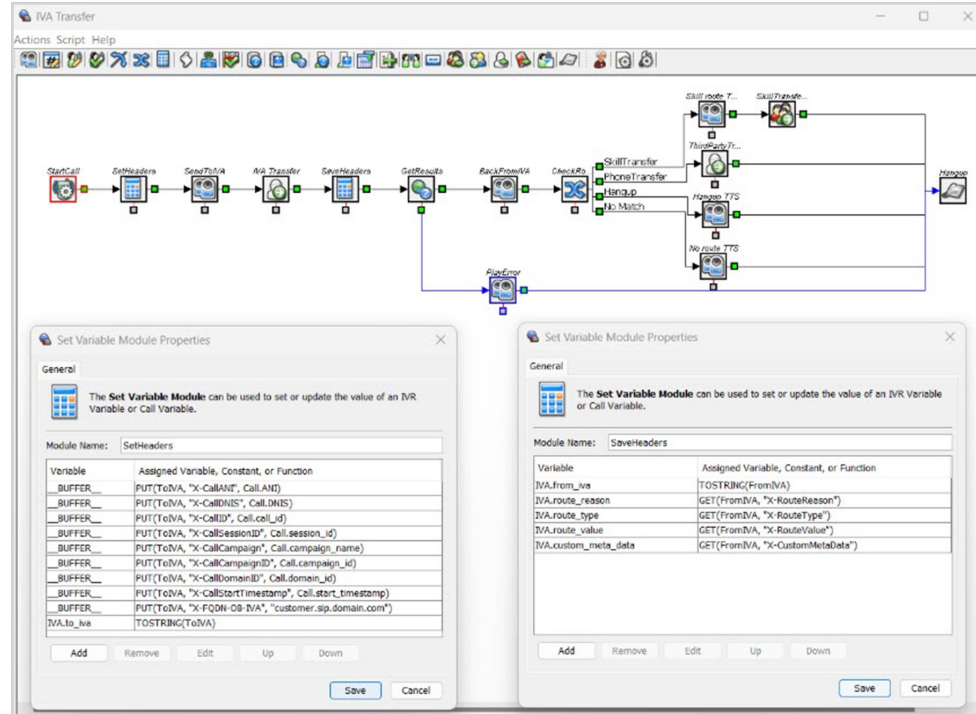
Variable	Type	Description
status_reason	String	Error reason
to_iva	String	Data sent to the IVA (KV list)



Note: The IVR script automatically populates these CAV values from corresponding **X-headers** in the API. For more information about headers, see [Header Usage](#).

3 Imports the **Sierra IVR Transfer Template v1** IVR script.

The following example shows an IVR transfer script.



Note: The Five9 implementation team uses the template designed specifically for Sierra, as **X-header** names vary across integrations.

The system already populates the production URL in the IVR template. The following example shows a pre-populated production URL in the IVR template.

The screenshot shows the 'Query Module Properties' dialog box. The 'General' tab is selected. The 'Module Name' is 'GetResults'. The 'Base URL path' is 'https://prod.us.com/five9/call-result'. The 'URL Port' is 'CallCallID'. The 'Comment' is 'Call ID'. The 'Method' is 'GET'. The 'Timeout (sec)' is '5'. The 'Save Number of Records to' is '5'. The 'Add', 'Remove', 'Edit', 'Up', and 'Down' buttons are visible. The 'Save' and 'Cancel' buttons are at the bottom right.

- 4 Configures the transfer number by using the single AI phone number (real) that Sierra provides for your organization. This configuration supports one phone number per customer, regardless of the number of domains. The Five9

implementation team sets the default value of the **iva_transfer_number** variable to this phone number.

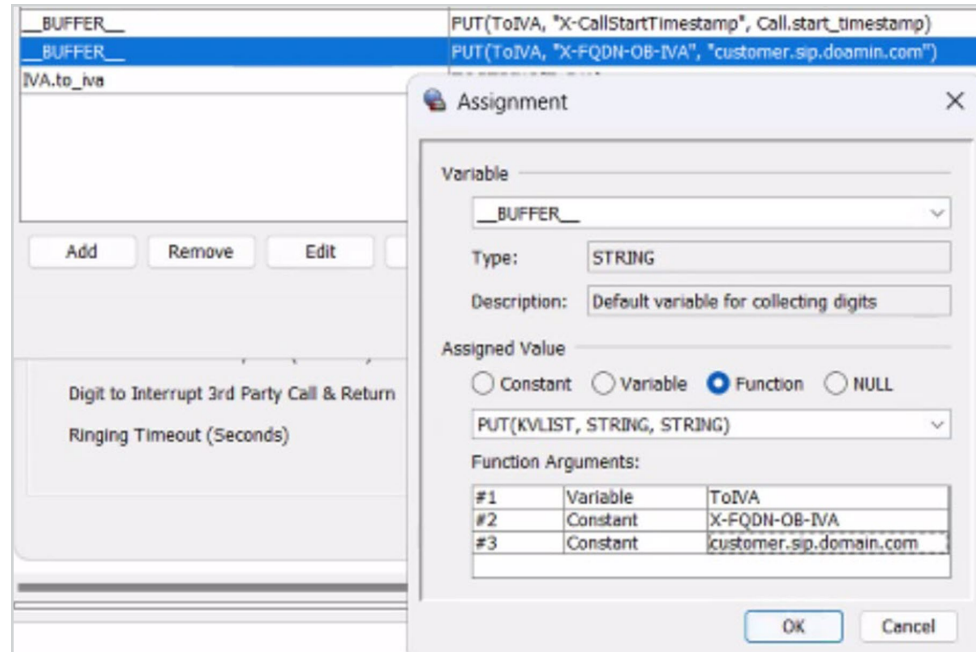
Note: The Connectivity PM manages call routing for the AI phone number to the IVA integration. The Five9 implementation team also uses the same transfer DID across all your domains that send calls to Sierra. If you use multiple AI agents, Sierra uses **X-headers** to differentiate routing within its platform.

The following example shows a transfer number configuration.

The screenshot shows the 'iva_transfer_number Properties' dialog box. The 'General' tab is selected. The variable name 'iva_transfer_number' is entered in the top field. Below it, the description is 'Set the Default Value of this CAV with the number provided by the Vendor'. There are two unchecked checkboxes: 'Reporting Call Variable (Saved in Database)' and 'Field Contains Sensitive Data'. The 'Data Type' is set to 'String'. Under the 'Restrictions' section, the 'Value' radio button is selected, and the value '9995550000' is entered in the text field. The 'Apply', 'Save', and 'Cancel' buttons are at the bottom right.

- 5 Sets the **X-FQDN-OB-PRD-SIERRA** header to the SIP domain value provided by Sierra. The Five9 implementation team uses the unique SIP domain assigned to your organization to ensure successful call routing through the Sierra integration.

The following example shows the unique SIP domain in the Set Variable module to enable call routing.



Header Usage

The Five9 implementation team manages **X-headers** in the IVA. The Five9 platform sends and receives **X-headers** during call processing, and the team can add more headers in either direction if needed. When adding custom headers, the team limits them to 10 to avoid packet fragmentation or call quality issues.

Headers Sent to the IVA

The Five9 platform sends the following platform call variables to the IVA:

- **call.ANI**
- **call.DNIS**
- **call.call_id**
- **call.session_id** (compare to snippet and update)
- **call.campaign_name**
- **call.campaign_id**

- **call.domain_id**
- **call.start_timestamp**

Note: The Five9 implementation team can edit the SetHeaders module to add more variables.

Headers Received from the IVA

The Five9 platform receives the following headers from the IVA:

- **X-RouteReason:** Provides a short outcome description for Five9 reporting purposes.
- **X-RouteType:** Specifies the next action the platform should take.

Example: Skill, external, hangup, agent

- **X-RouteValue:** Identifies the destination for the next action.

Example: Skill name, phone number, other routing value

- **X-CustomMetaData:** Returns supplemental data such as breadcrumbs, short summaries, or other contextual information required for reporting or troubleshooting.

Note: The Five9 implementation team can edit the SetHeaders module to add more headers.