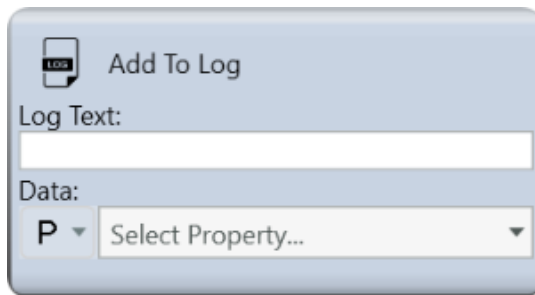


# ECS Setup: Interaction Handling Flow: Data Set and Get Nodes

- Add To Log
- Add Remark
- Add to DNC
- Block Until Contact is Loaded
- Clear Demand
- Format Text
- Get Contact History Data
- Get Current Date Time
- Get Email Data
- Get External Data by REST API
- Get Queued Interaction Statistics
- Get US State
- Increment
- Load CRM Customer Information
- Load Customer Information
- Search Key Words
- Set Campaign Final State
- Set Data
- Set Demand
- Set Demand by Variable
- Set Destination Address
- Set Disposition Code
- String Manipulation

## Add To Log



The screenshot shows a configuration window for the 'Add To Log' activity. It has a title bar with a 'Log' icon and the text 'Add To Log'. Below the title bar, there is a 'Log Text:' label followed by a text input field. Underneath that is a 'Data:' label followed by a dropdown menu. The dropdown menu has a 'P' icon on the left and the text 'Select Property...' in the center.

**Media Type:** All

**Purpose:** This activity is used for troubleshooting purposes. When configured into the interaction handling flow it will automatically add a log entry within ECS application logs.

**Body Attributes:**

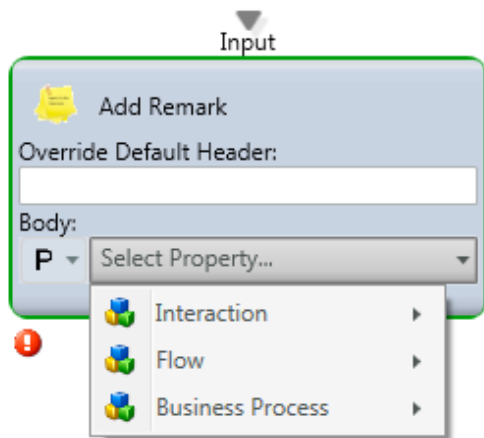
- Constant Expression is static information
- Property Expression is dynamic information based on Interaction, Flow or Business Process attributes.

**Output Legs:** N/A

**Common Use Case:**

This activity should only be used under the guidance of Evolve IP support.

## Add Remark



**Media Type: All**

**Purpose:** Automatically add a Note/Remark that is passed down to the agent along with the interaction. These remarks are also available in reporting via analytics and can also be used for troubleshooting purposes when changing call flows.

**Body Attributes:**

- Constant Expression is static information
- Property Expression is dynamic information based on Interaction, Flow or Business Process attributes.

**Output Legs: N/A**

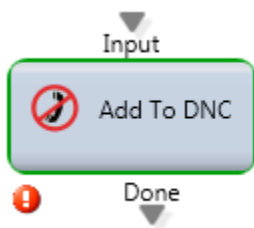
**Common Use Case:**

Providing Agent Information: The agent can be provided multiple remarks during the call flow which will be presented to them in the "Remarks" section of the agent client. This is useful for providing the agent information about the prompts selected by the caller (such as "billing question") or data captured from external systems (such as "account number") or other types of information (such as the marketing campaign or location contacted) so the agent knows how to treat this individual.

**ITaaS Submenu: Service Issue: Desktop,  
ITaaS: ITaaS After Hours Call**

Troubleshooting: Remarks can also be used for troubleshooting new call flow. You can add remarks within the call flow to show the direction a call took at specific times in the routing.

## Add to DNC



**Media Type: Voice**

**Purpose:** Automatically add the caller's phone number to the "Do Not Call" list. This information is stored on the Database of the customer's organization

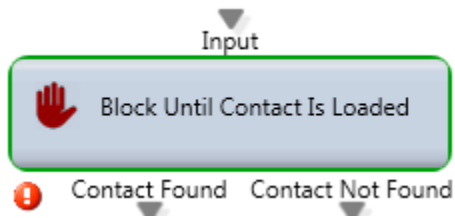
**Attributes: None**

**Output Legs: N/A**

**Common Use Case:**

The most common use case is in outbound dialer scenarios where the caller can opt into the "Do Not Call" list to no longer receive campaign calls. For example, press 1 to connect to an agent, press 2 to be added to the DNC list. The origin number is captured and added to the list.

## **Block Until Contact is Loaded**



**Media Type:** All

**Purpose:** Pause the interaction handling until the requested record is returned from the internal CRM system.

**Attributes:** N/A

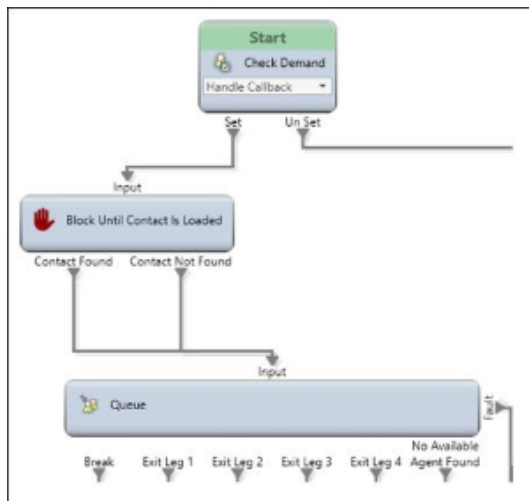
**Output Legs:**

- **Contact Found** – path to follow if the contact is found.
- **Contact Not Found** – path to follow if the contact is not found.

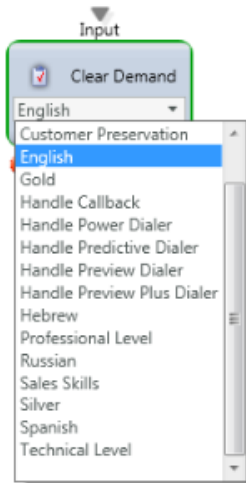
**Common Use Case:**

This will not pass the interaction until the contact information is searched in the internal CRM. This node is commonly used in scenarios where the Contact information is needed before the interaction continues. Common scenarios for this Activity include:

1. Retrieving the Contact record before presenting an Agent with a Callback.
2. Making routing decisions based upon information stored in the Contact record.



## **Clear Demand**



**Media Type:** All

**Purpose:** Deactivate a previously activated Agent Skill demand

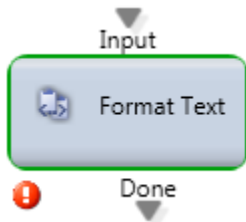
**Attributes:** List of available demand(s)

**Output Legs:** N/A

**Common Use Case:**

This node is used to clear a demand that was set through an interaction flow. Some common uses of this node include clearing a demand when a specific queue threshold is reached or in after-hours routing when your agents skilled in a particular demand are not staffed.

## Format Text



**Media Type:** All

**Purpose:** Formats the text of the URL that will be visible on the agent client and used to launch a 3rd party application via their browser.

**Attributes:**

Template	Target
SFDC Object	Interaction > Product Link

Formatting Values
<div> <div>Object ID</div> <div>P</div> <div>Interaction</div> <div>Contact</div> <div>Text 12</div> </div>

**Output Legs:** N/A

**Common Use Case:**

Present the agent with a dynamic or static web link based on customer input or other relevant information. The Format Text activity combines the link template created in the Text Template Repository with the actual interaction's values and assigns that link to the intended "Target". There are 3 potential link Targets (Customer Management, Location, and Product) that can be visible on the agent client.

**Example:**

Template = "SFDC Object" which is defined as "[https://login.salesforce.com/<Object\\_ID>](https://login.salesforce.com/<Object_ID>)" in the Text Template Repository

Location = Product Link

Formatting Values – this is where the <Object\_ID> placeholder is replaced with the actual value for the interaction. In this example, the value held in the Text 12 field will be inserted for the <Object ID> in the URL made available in the Agent client.

## **Get Contact History Data**

**Media Type:** All

**Purpose:** Retrieves the details (Origin, Destination, Status, Last Agent, Last Disposition, Last Interaction Type, and Last Business Process) of this customer's last interaction to write to interaction flow variables.

**Attributes:**

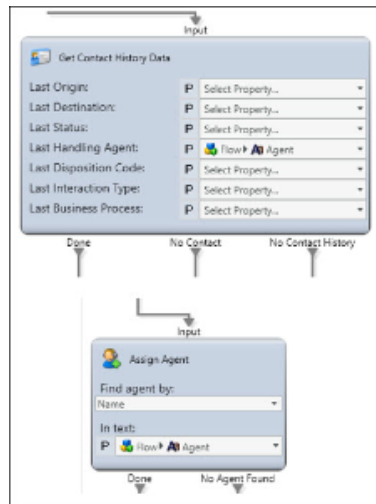
- **Last Origin:** The phone number or email address associated with the last interaction
- **Last Destination:** The last number (voice) or business process (chat and email)
- **Last Status:** The final status of the last interaction (Handled, Callback Requested, Abandoned, etc.)
- **Last Handling Agent:** The agent that handled the last interaction (FirstName LastName)
- **Last Disposition Code:** The Disposition Code that was assigned to the last interaction
- **Last Interaction Type:** The type of the last interaction (Telephony, Chat, Email, etc.)
- **Last Business Process:** The business process associated with the last interaction

**Output Legs:**

- **Done:** The information is obtained and populated to the assigned variables
- **No Contact:** There are no stored contacts in the CRM for this interaction
- **No Contact History:** There is a contact in the CRM, however this is their first interaction

**Common Use Case:**

Route interaction to last handling agent: In this scenario you can obtain the last handling agent and write this information to a flow variable "Agent". You can then use the Assign Agent node with attribute "Name" selected for your "Agent Flow Variable" (Pictured below)

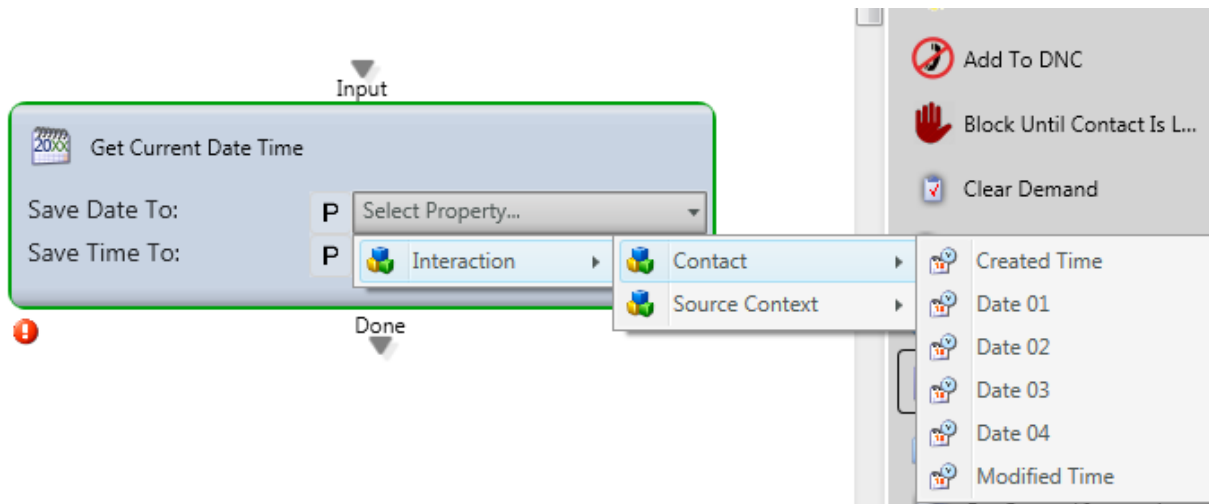


You could then send this interaction to a queue that will attempt to deliver the interaction to the assigned agent. Typically inside the queue, the following value would be configured to denote the amount of time to wait for that specific agent before delivering that interaction to any Ready agent.

**Time To Clear Assigned Agent**

0 sec

## Get Current Date Time



**Media Type:** All

**Purpose:** Retrieves the current date and time

**Attributes:** Save the current date and time to the desired property.

**Output Legs:** N/A

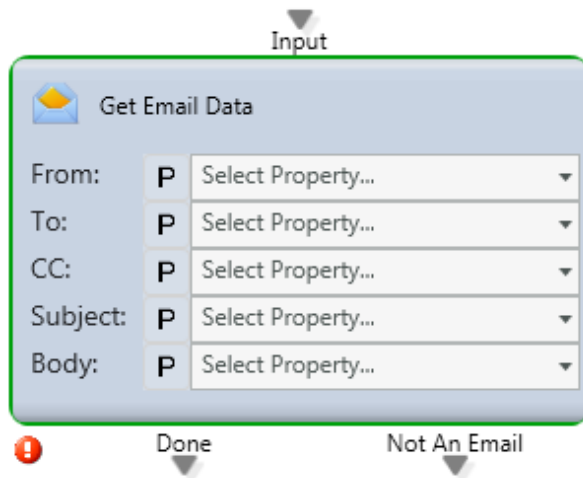
**Common Use Case:**

This node is used to store the current interaction's date and time so it can be used in the interaction handling flow.

**Example:**

Time and date of the interaction is stored and compared to other values in the interaction handling flow to determine the proper routing logic. For example, a Branch activity could be used to compare the current date to an expected payment date in the CRM to see if the customer's payment is considered "late" so the interaction can be routed to the appropriate Business Process.

## Get Email Data



**Media Type:** Email

**Purpose:** Ability to assign Email Header data elements (From, To, Subject, CC, Body) to Properties

**Attributes:**

- **From:** The Property to store "From" section of the email
- **To:** The Property to store the "To" section of the email
- **CC:** The Property to store the "CC" section of the email
- **Subject:** The Property to store the "Subject" section of the email
- **Body:** The Property to store the "Body" section of the email

**Output Legs:**

- **Done** – the values were stored to the specified Properties
- **Not An Email** – the interaction is not an email and these values were not stored

**Common Use Case:**

The common use case for this node is to obtain information from the email and take specific action/routing based on keywords within the selected variable. You can use the [Search Keywords](#) node and route to a specific Business Process based on certain keywords within the subject or body for example.

**Example**

An email arrives regarding an inquiry from your company's latest social media post. You are currently tagging the body of the email to a flow variable "Body" via the [Get Email Data](#) node. Using the [Search Keywords](#) node you search the assigned "Body" flow variable for specific words, in this case "Twitter". Based upon the keyword match you route the call to your customer service agents while also tagging that interaction with a skill of "Marketing" so to present the interaction to a properly skilled agent.

## Get External Data by REST API



**Media Type:** All

**Purpose:** Retrieve data from an external CRM or business application.

**Attributes:**

- **Rest API Data:** Select the appropriate REST API Data method. Note: the method is configured in the Implementation: REST API.

#### Output Legs:

- **Done** – the values were stored to the specified Properties
- **Fault** – a fatal error occurred

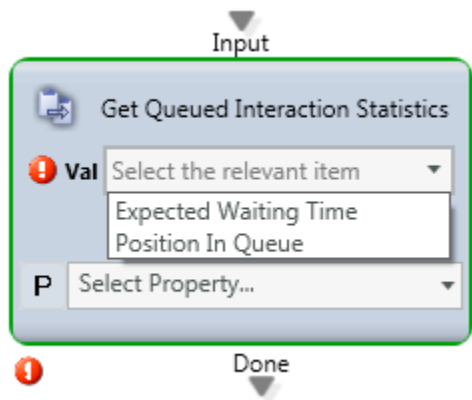
#### Common Use Case:

The common use case for this node is to retrieve information about the customer that can be used to intelligently route the interaction, present updated information to the customer, offer relevant prompts, and deliver key customer information to the Agent.

#### Example

An incoming call is received and the caller's Origin DID is passed to the external system to identify the account associated with the caller. The caller's information is leveraged to determine the appropriate destination and related prompts. The caller's account information is passed to the Agent who immediately knows the caller's identity.

## Get Queued Interaction Statistics



**Media Type:** All

**Purpose:** Allows the assignment of Position in Queue or Expected Waiting Time to a Property

#### Attributes:

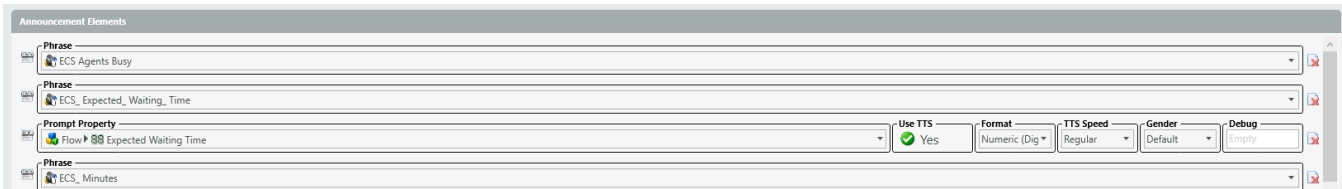
- **Expected Waiting Time:** This will assign the Expected Waiting Time to a flow variable (Flow Type must be Count)
- **Position In Queue:** This will assign the Position in Queue to a flow variable (Flow Type must be Count)

ExpectedWaitingTime	Count
---------------------	-------

**Output Legs:** N/A

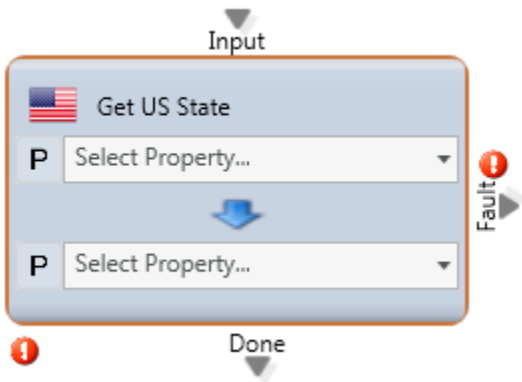
#### Common Use Case:

In order to be able to read back the callers position in queue or expected waiting time this node must be used. You will obtain the value and write to a Flow Variable which can then be played back via a Simple Play node.



## Get US State





**Media Type:** All

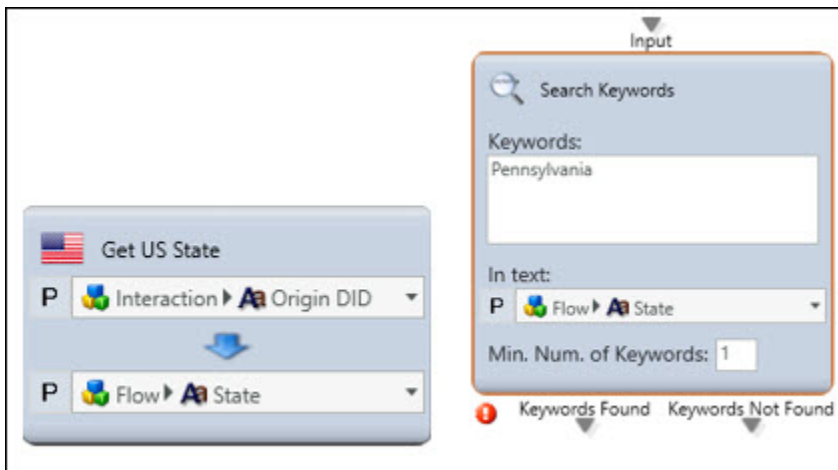
**Purpose:** Allows the assignment of State to a Property

**Attributes:** Drop down for Interaction, Flow Variables or Business Process fields. The most common used Property is "Interaction -> Origin DID".

**Output Legs:** N/A

**Common Use Case:**

The common use case for this node is the ability to route based on incoming caller area code. **Get US State** along with **Search Keywords** will allow you to achieve this routing. For instance let's say you have a business process for 4 specific regions (North, South, East and West). You would like to route the caller based on their incoming number area code to a specific region for promotional purposes. You would accomplish this by setting the Interaction > Origin DID to a Flow Variable, in this case called "US State"



This will return the full state name based on the caller's area code I.E 610 would return "Pennsylvania") You can then search keywords for Pennsylvania to route to your East Business Process. This is also useful when certain states require specific call recording guidelines to which you can identify and treat those callers differently.

## **Increment**

**Media Type:** All

**Purpose:** Increments or Reduce a Property value to enable different Branch destinations during the interaction flow.

**Attributes:**

- **Add:** This will add the selected amount to the variable you select in the To/From dropdown.
- **Subtract:** This will subtract the selected amount to the variable you select in the To/From dropdown.
- **Amount** – The amount you would like to add or subtract from the selected property value.

**Output Legs:** N/A

**Common Use Case:**

The common use case for the increment node is to verify a specific property value and route based on whether that number is "> = <" your expected result.

## Load CRM Customer Information

**Media Type:** All

**Purpose:** Routing decision based upon the lookup of interaction information in the internal CRM. The system will automatically route based on contact Phone1 or Email without the need to input this node.

**Attributes:**

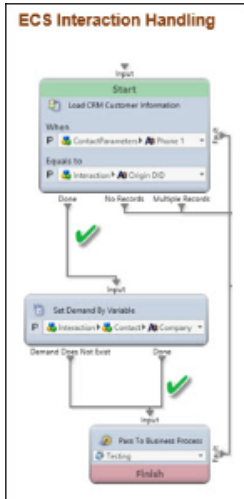
- **"When" Contact Parameters:** These are the specific parameters within the CRM. Phone 1, Company, First Name, Last Name etc. as well as custom text fields Text01, Text02 etc.
- **"Equals To" Interaction, Flow or Business Process Variables** – These are variables that are passed via interaction I.E (Interaction > Origin DID) or a contact match I.E. (Interaction> Contact >Company) or within selections from the call flow (Flow Variables).

### Output Legs:

- **Done:** The path to follow once the information is properly matched
- **No Records:** The path to follow if no information is matched
- **Multiple Records:** The path to follow if multiple records matched

### Common Use Case:

You can perform custom routing based on fields uploaded within the CRM. For instance, When **Contact Parameters > Phone 1** equals to **Interaction > Origin DID** assign the company as a skill demand. Those interactions can now be answered by those agents who are proficient with that company



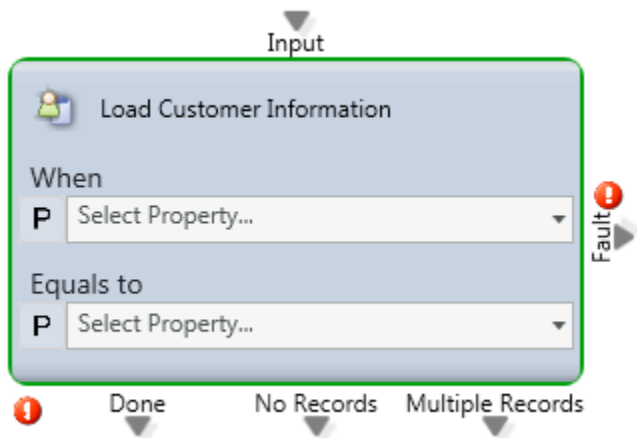
**Agent Popup**

Agent 2 00:00:01 3:37 PM

6105291714 00:00:01

	Contact	Search	History	Notes
State	Handling			
Type	Incoming Telephony	Primary Phone		
Customer		Company	Evolve IP	✓
Entry Time	3:37 PM	First Name	Rich	
Origin		Last Name	Cannon	
Destination	6102637555			
Business Process	Testing			
Demands	Evolve IP			✓
Waiting Time	00:00:00			
Remarks	0			
Links				
Previous Attempts	0 previous attempts			
Completion Status	Handled			
123	Telephony Sessions			

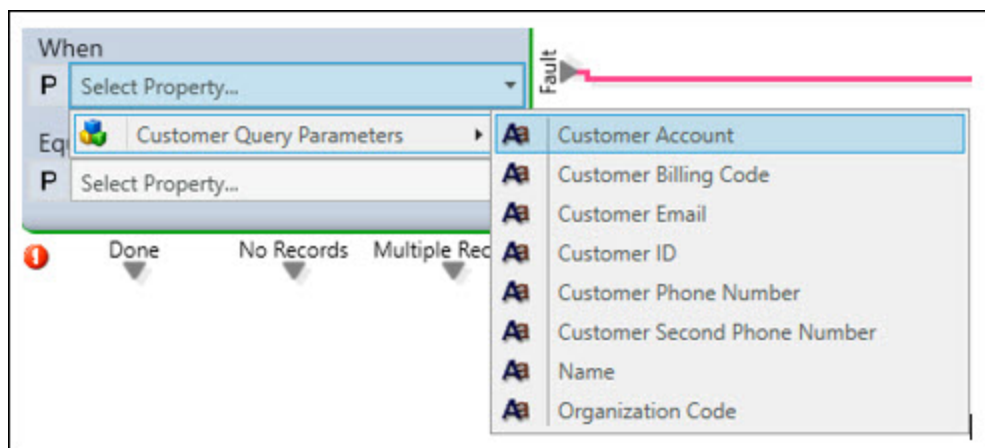
### Load Customer Information



**Media Type:** All

**Purpose:** Routing decision based upon the lookup of the Customer Data upload via the application portal

**Attributes:** Customer Data template information

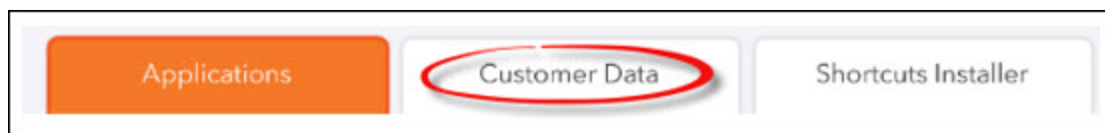


#### Output Legs:

- **Done:** The path to follow once the information is properly matched
- **No Records:** The path to follow if no information is matched
- **Multiple Records:** The path to follow if multiple records matched

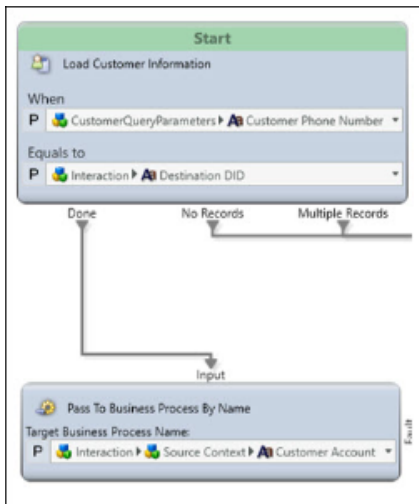
#### Common Use Case:

This node can be used when you want to route a specific number (origin or destination) to a particular Business Process. You can also add in specific demands/skills for those interactions. The data set for this information is stored in "Customer Data" tab of the application portal.



In this example, any calls to 610-263-4322 will be routed to the Testing Queue.

	A	B	C	D	E
1	CustomerID	CustomerEmail	CustomerPhoneNumber	CustomerSecondPhoneNumber	CustomerAccount
2	1		6102634322		Testing Queue



## Search Key Words

The 'Search Keywords' window is shown with an 'Input' arrow at the top. It contains a 'Keywords:' text area, an 'In text:' dropdown menu with 'Select Property...' selected, and a 'Min. Num. of Keywords:' input field set to '1'. At the bottom, there are two output paths: 'Keywords Found' (marked with a red exclamation mark icon) and 'Keywords Not Found'.

**Media Type:** All

**Purpose:** Routing decision based upon 1 or more text keywords found in the interaction.

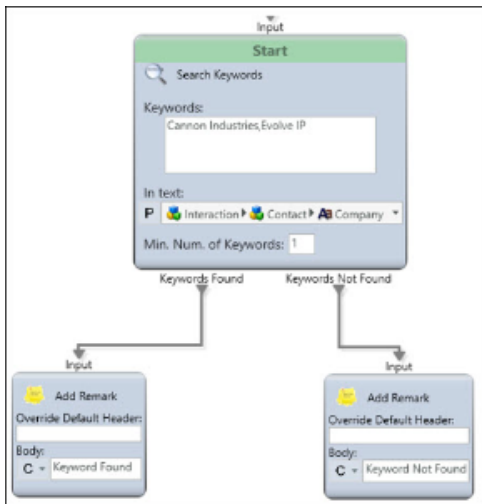
**Attributes:** Interaction, Flow or Channel variables.

**Output Legs:**

- **Keywords Found:** This path will be activated if the specified keywords are found.
- **Keywords Not Found:** This path will be activated if the specified keywords are not found

**Common Use Case:**

Common use case referenced above in the [Get US State](#) node. You can also route callers based on keywords such as the Company name. Multiple keywords should be separated by commas.



The screenshot shows the Agent interface for a call with ID 6105291714. The call is in 'Handling' state and is an 'Incoming Telephony' call. The 'Contact' tab is active, showing fields for 'Primary Phone', 'Company' (Cannon Industries), 'First Name' (Rich), and 'Last Name' (Cannon). The 'Remarks' field shows a list of remarks, including 'Carrier 10/22/2018 3:17:44 PM' and 'Keyword Found'. The 'Completion Status' is 'Handled'. The 'Telephony Sessions' dropdown is set to '123'. The 'Add Remark' button is visible at the bottom.

## Set Campaign Final State

The screenshot shows the 'Set Campaign Final State' dialog box. It has a dropdown menu for 'Customer Reached' and a list of options: 'Answering Machine', 'Busy Tone', 'Customer Reached' (highlighted), 'Fax', 'No Answer', 'Service Unavailable', 'Unknown Error', and 'Wrong Number'.

**Media Type:** Voice

**Purpose:** To set the final state of a campaign dialer call.

**Attributes:**

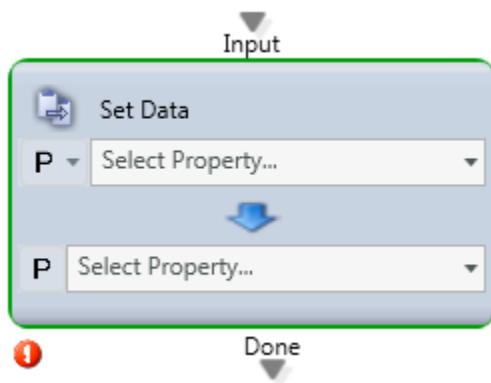
- Answering Machine
- Busy Tone
- Customer Reached
- Fax
- No Answer
- Service Unavailable
- Unknown Number
- Wrong Number

**Output Legs:** N/A

**Common Use Case:**

This is used when you want to specify the final state of a particular call path used within a dialer. This is also commonly used with Set Disposition Code to properly tag the call for reporting purposes.

## Set Data



**Media Type:** All

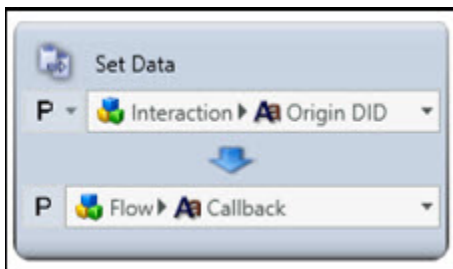
**Purpose:** Assign data to a specific Property

**Attributes:** Interaction, Flow and Business Process property variables

**Output Legs:** N/A

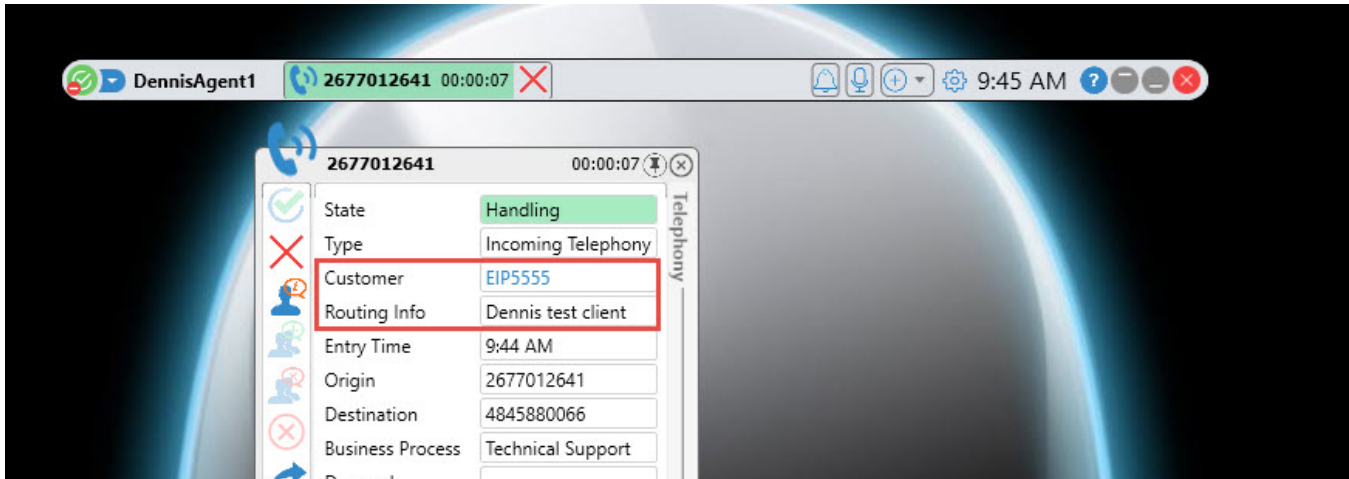
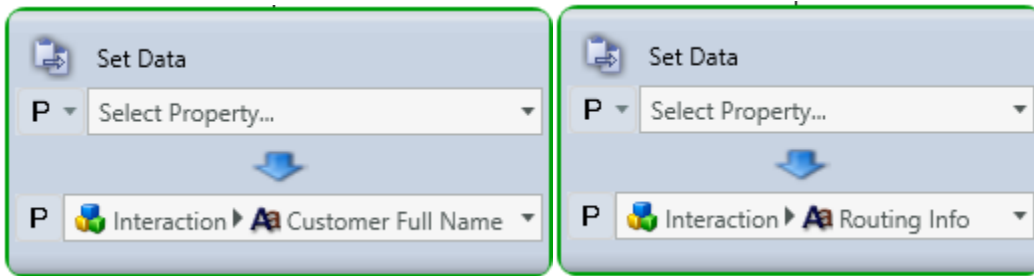
**Common Use Case:**

Taking interaction data and writing to a specific variable such as a flow variable. This is commonly used in Callback flows where first you set a flow variable to the originating DID of the caller.

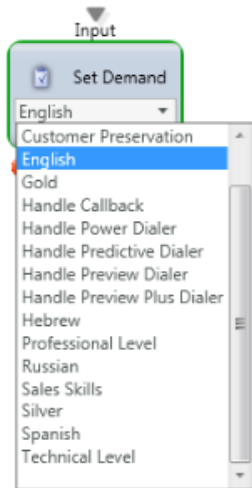


You could then read back the **Flow > Callback** number to the caller and verify if they want to keep or change. If they would like to modify the callback number you can change the **Flow > Callback** number in a Prompt and Collect Data node.

Another option is to pull customer specific information through Interaction variables. As an example, the **Customer Full Name** and **Routing Info** variables will populate the **Customer** and **Routing Info** fields within the agent Client.



## Set Demand



**Media Type:** All

**Purpose:** Establishes a specific Agent Skill that is required for this interaction

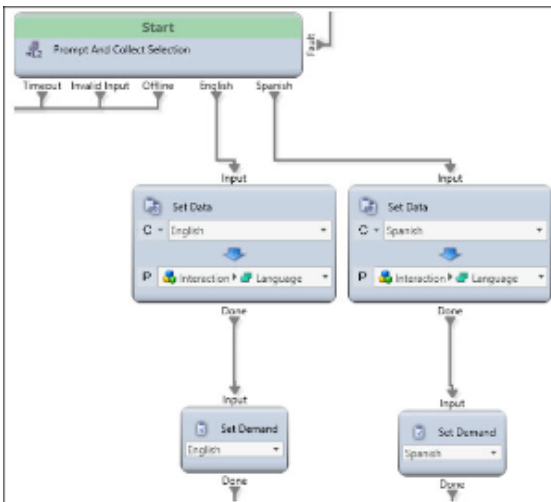
**Attributes:** A dropdown list of demands which are specified in **Implementation > Skills**.

**Output Legs:** N/A

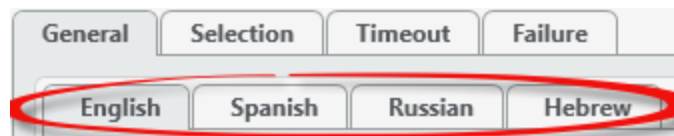
**Common Use Case:**

Demands are used within skill based routing and will carry over throughout the Interaction unless a [Clear Demand](#) or [Clear All Demands](#) node is used. Your agents are skilled in particular demands via **Staffing > Agent Characteristics** section. Please keep in mind that multiple demands can be assigned to a single interaction.

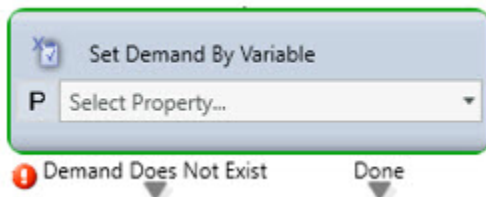




The most common example of skills based demands is language selection. If you have a [Prompt and Collect Selection](#) where you ask the end user to select 1 for English or 2 for Spanish, you will then set the language requirement via Set Data as well as the demand before routing the interaction. The Set Data language setting is needed to correctly play the appropriate language based greetings in all Announce and Collect nodes.



## Set Demand by Variable



**Media Type:** All

**Purpose:** To dynamically assign a demand based on a property value

**Attributes:** Interaction, Flow and Business process property variables.

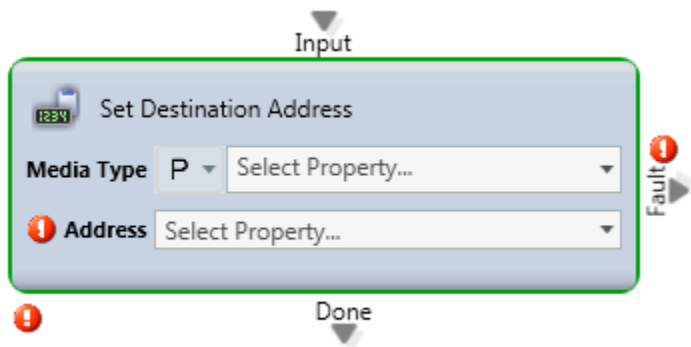
**Output Legs:**

- **Demand Does Not Exist** – path followed when the Property value does not match a valid Skill
- **Done** – path followed when the Property value matched a valid Skill.

**Common Use Case:**

This is commonly used in conjunction with the [Customer Data](#) upload. Where you can set a dynamic demand on a per caller/calling number basis. This was also shown before in the [Load CRM Customer Information](#) where we dynamically set the demand to the callers **Interaction > Company** so that they will be answered by an agent skilled for their particular account.

## Set Destination Address



**Media Type:** Email, Fax, Voice

**Purpose:** Establishes the phone number used to call back a customer

**Attributes:**

- **Property** – Interaction, Flow or Business Process Property values
- **Contact** – Email, Fax, Voice

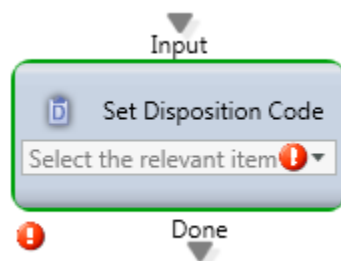
**Output Legs:** N/A

**Common Use Case:**

The most common use case is for the setup of a voice callback. In this case you would set the media type as Constant and Select Telephony. The address section would then be populated with your flow variable **Flow > Callback**.



## Set Disposition Code



**Media Type:** All

**Purpose:** Automatically assign a Disposition Code to an interaction

**Attributes:** A dropdown list of disposition codes. The list is determined by the Disposition Code Set assigned to the Business Process.

**Output Legs:** N/A

**Common Use Case:**

This can be used anywhere in the flow to assign a disposition code to an interaction for reporting purposes. The agent will ultimately be able to modify this Disposition Code in the Agent client when handling the interaction.

## String Manipulation



**Media Type:** All

**Purpose:** The ability to find and filter specific characters in a flow variable text string.

**Attributes:** N/A

**Output Legs:** N/A

**Common Use Case:**

**Action Types:**

- **Find Index of String** – This will return the position of a specific character within the selected input string.
- **Sub String** – This will return the text of selected input string from the specified starting position and character length

Find Index of String Example Using Chat Requestor Property – we will find the position of the @ symbol in this string.

**Input:** **Interaction > Origin > Address Field > URL** which returns the Chat requester's information in the format of "First name, Last Name (email@domain.com)"

**Text to Find:** We are looking in this text string for the "@" symbol in the email address.

**String Output:** We are outputting this value to flow variable **Flow > Domain Index**

**Position Output:** We will store this position in **Interaction > Source Context > Age**

<b>Action Type</b> Find Index Of String	<b>Input</b> P - Interaction > Origin > Address Field > URL	<b>Start Position</b> P - Select Property...
<b>Length</b> P - Select Property...	<b>Text To Find</b> C - @	
<b>String Output</b> P - Flow > Domain Index	<b>Position Output</b> P - Interaction > Source Context > Age	

Sub String Example Using Chat Requestor Property – we will capture the email domain name including the @ symbol.

We will now use another String Manipulation node this time with Action Type: Sub String

**Input:** We will use the same input of **Interaction > Origin > Address Field > URL**

**Start Position:** This time we will start looking at the position we saved in the step above in **Interaction > Source Context > Age** which is the position of the "@" symbol

**Length:** A length of 0 indicates the remaining part of the string will be returned.

**String Output:** We will write this value to a new flow variable **Flow > Domain Sub String**

<b>Action Type</b> Sub String	<b>Input</b> P - Interaction > Origin > Address Field > URL	<b>Start Position</b> P - Interaction > Source Context > Age
<b>Length</b> C - 0	<b>Text To Find</b> P - Select Property...	
<b>String Output</b> P - Flow > Domain Sub String	<b>Position Output</b> P - Select Property...	

Using the Remarks activity, we display the Sub String and complete Chat Requester in the Agent client below.

Rich Cannon (rcannon@evolveip.net)

✓

✗

👤

👤

✗

➡

State

Type

Customer

Entry Time

Origin

Destination

Business Process

Demands

Waiting Time

Remarks

Links

Previous Attempts

Completion Status

Handling

Incoming Chat

✓

Rich Cannon

(rcannon@evolveip.net)

4:02 PM

Rich Cannon

(rcannon@evolveip.net)

Channel

Testing

00:00:00

2

0 previous attempts

Handled

Contact

Search

History

Notes

Primary Phone

First Name

Last Name

Company

Email

Rich

Cannon

evolve ip

rcannon@evolveip.net

SubString

10/23/2018 4:02:15 PM

@evolveip.net

✓

Interaction URI

10/23/2018 4:02:16 PM

Rich Cannon (rcannon@evolveip.net)

✓

Write Here...

00:00:35

Chat With: Rich Cannon (rcannon@evolveip.net)

Write Here...

Submit