Simplify. Track. Improve.

Live.CCS

Custom Call popups combined with advanced analytics.
The service you need to regain control over your support call center.

IVR Setup Guide

Contents

Introduction

Live.CCS is a solution for Support Call Center analytics and a Call Popups provider. You can get additional information on the web: **https://solutions.si/ccs**

This guide will walk you through the integration of 3CX Call Flow with Live.CCS. Live.CCS integrates with IVR Flow in 4 steps. The guide is broken down into sections, each describing one of the steps. To successfully integrate the systems, you must add all 4 components into various regions of your Call Flow.

The guide presumes you have basic or no experience with 3CX Call Flow Designer. The level of experience at the beginning starts at very low and covers all needed concepts for further steps. In case you don't see some labels included in the document, make sure you revisit previous sections and fix your mistakes.

In case you encounter any trouble during the implementation, do not hesitate to ask us. The best way to reach us is by email: **info@solutions.si**

Pre-setup step

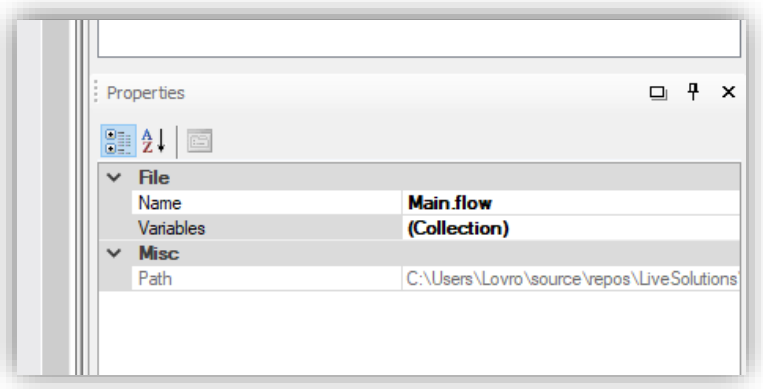
Setting the API key

Live.CCS system uses API keys for user authentication. API key is a short sequence of letters and numbers, which uniquely identifies users and IVR Flows. Each IVR Flow represents a separate user. We will issue one API key for each of your IVR Flows.

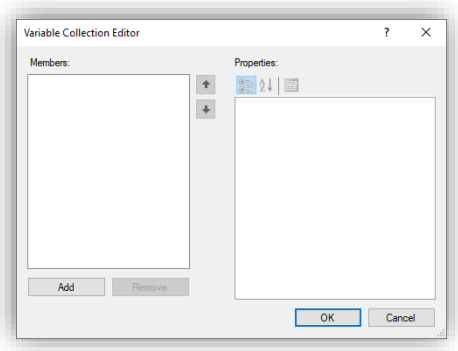
For example: if you have English, Spanish and German support lines, you will receive 3 different API keys, one for each Flow.

You must save the API key as a global variable in each Flow.

1. Open the main flow, click somewhere on whitespace of the workspace, and observe the properties:

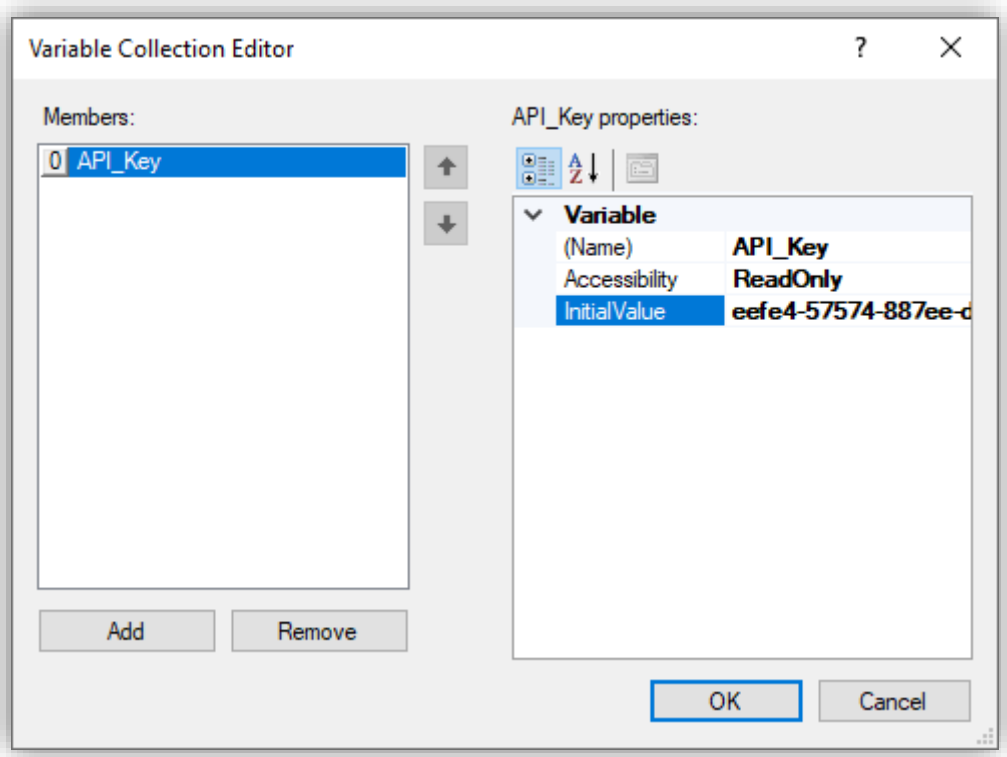


2. Click on the "**(Collection)**" cell of the "**Variables**" row and select the ellipsis button (**ction)** ...) at the end. A window should appear containing your existing variables:



3. Click on the "**Add**" button. A new row will appear on the left. In the properties panel on the right, edit the values as follows:

    - (Name) = "**API_Key**"
    - Accessibility = "ReadOnly"
    - InitialValue = **<API KEY OF THIS SPECIFIC FLOW>**



Then Press "**OK**".

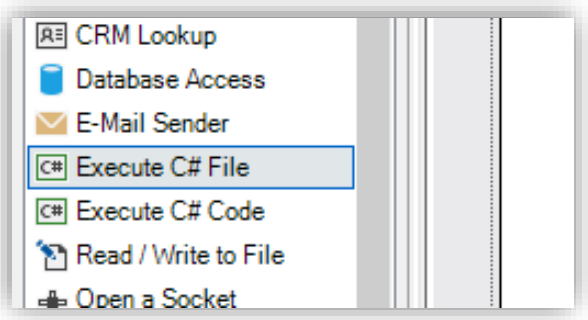
Component #1

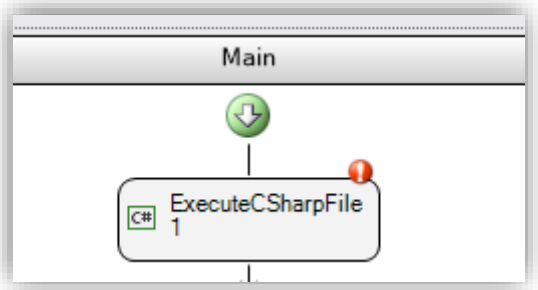
Session key generator

Each flow run represents a **session** identified by a UUID identifier (GUID). Each flow run generates its own session number which is also used for event grouping later in the process.

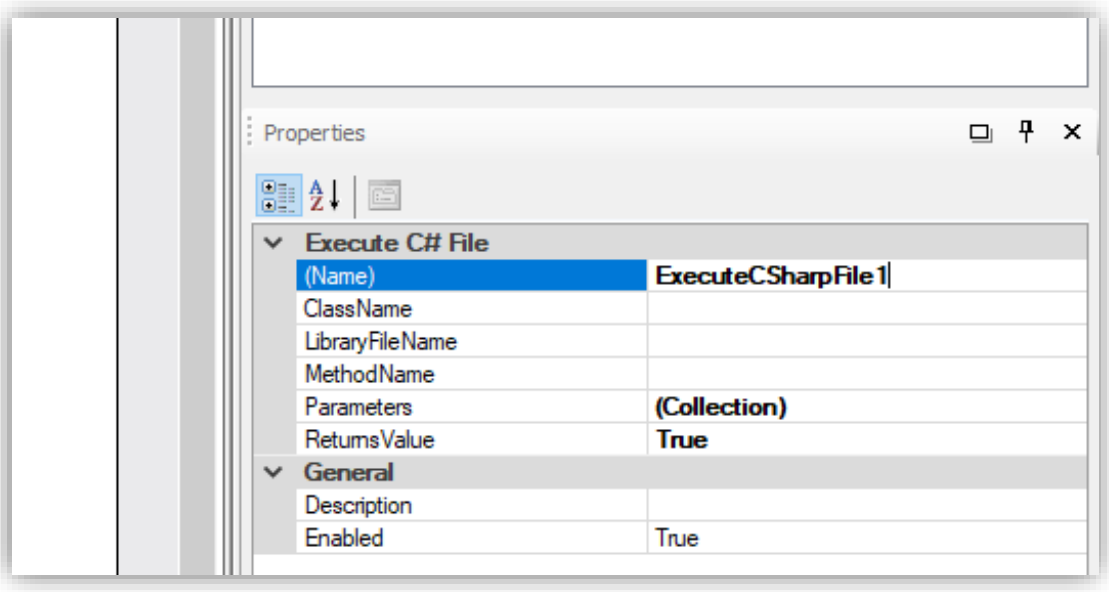
Session generator must be placed right at the beginning of the flow:

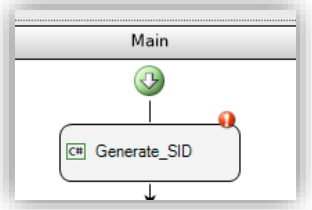
1 Add »Execute C# File« step:





2 Rename the step for easier identification later in the flow. Use »**Generate_SID**« as the name

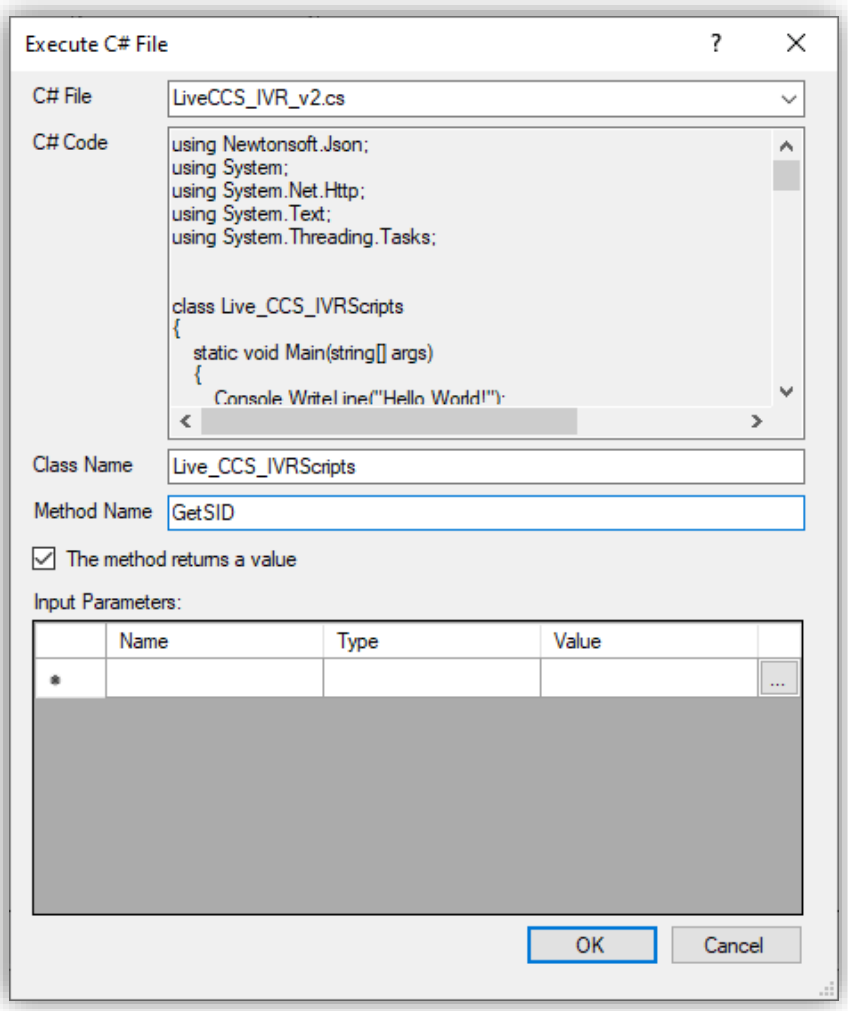




3 Double click on »**Generate_SID**« block. In the window, choose the file »**LiveCCS_IVR_v2.cs**« provided along with this manual. The »C# code« textarea should populate with the C# code from the file.

4 Put the following values in the fields:

1. Class Name = »**Live_CCS_IVRScripts**«
2. Method Name = »**GetSID**«
3. The method returns a value = ☑



5 Press »**OK**«

Component #2

Call-receive notification

Phone call represents the entry event for any session. The process of sending the Call-receive notification is like generating a session.

1 Add a new "**Execute C# File**" step **after** the session generator step (added in the previous section).

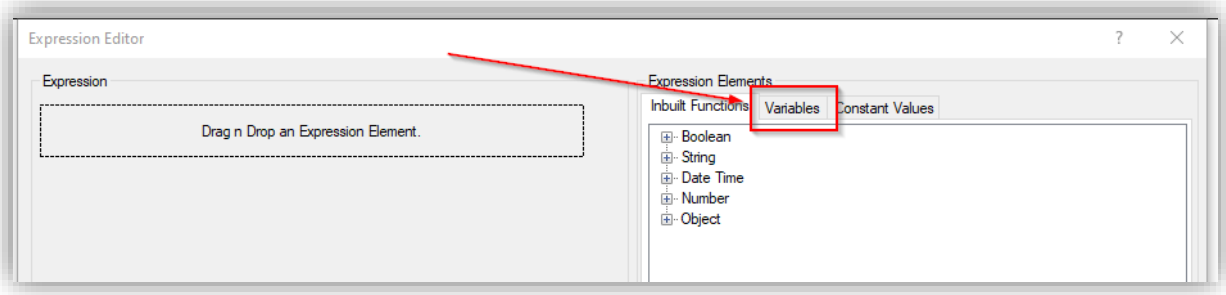
2 Double click the new step. In the "**C# File**" dropdown, again, select the "**LiveCCS_IVR_v2.cs**" file. The "C# Code" field should now contain the code.

3 Enter the following data in the fields:

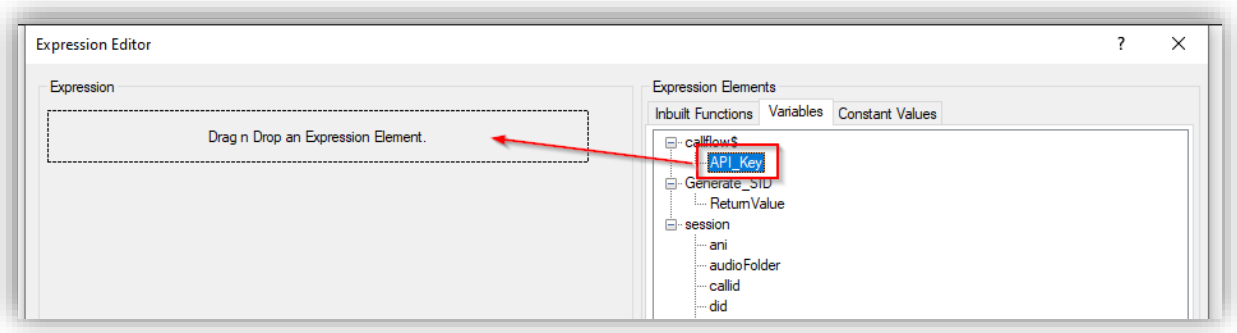
- Class Name" = »**Live_CCS_IVRScripts**«
- "Method Name" = "**SignalUserCall**"
- "This method returns a value" = ☐

4. In the table at the bottom labeled "**Input parameters**", click inside the cell below "Name" header and enter "**APIKey**". In the Type selector, select "**string**" instead of Boolean and then click on the ellipsis button after the Value cell.

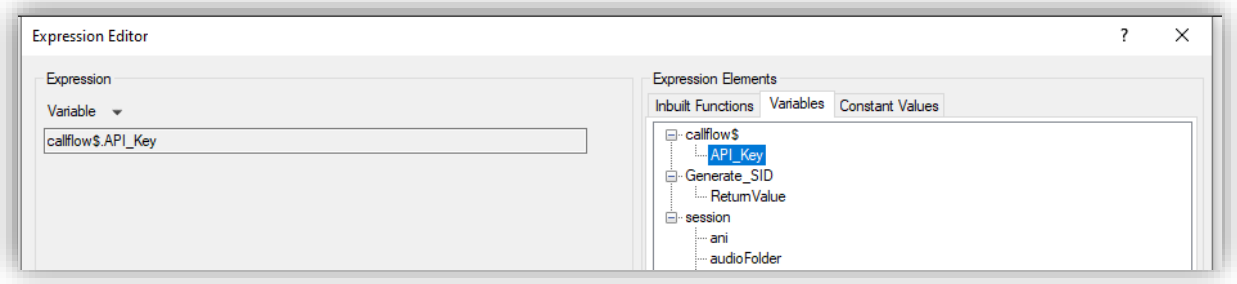
5. In the window which appears, select "Variables" tab



6. Click the "**API_Key**" label on the right and **DRAG and DROP** it to the left window labeled "**Drag n Drop Expression Element**"

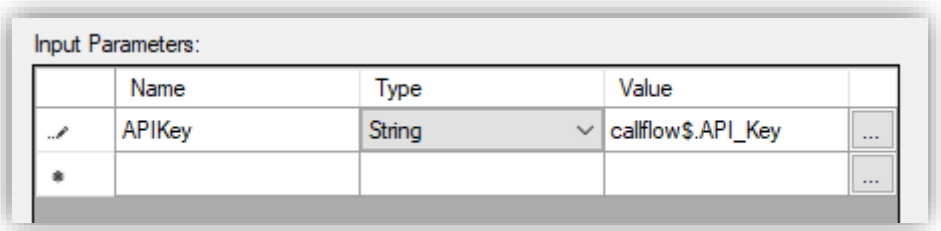


The window should look like this after you have dropped the value over to the left:



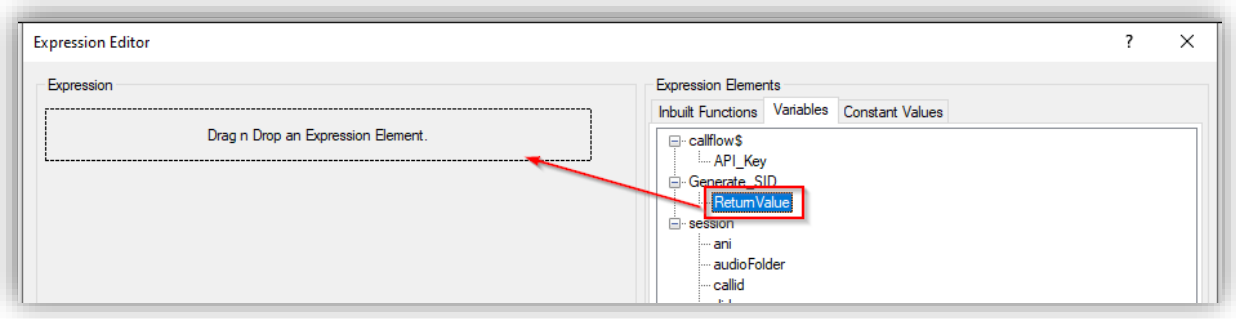
7. Select "OK" at the bottom

8. Right now, the "Input parameters" should look like this:

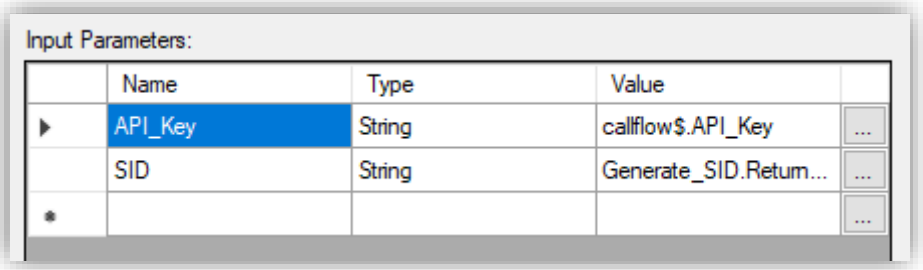


9. Click on the empty cell below APIKey and enter "**SID**". Select "String" as the type and again, click on ellipsis on the newly created row (SID row).

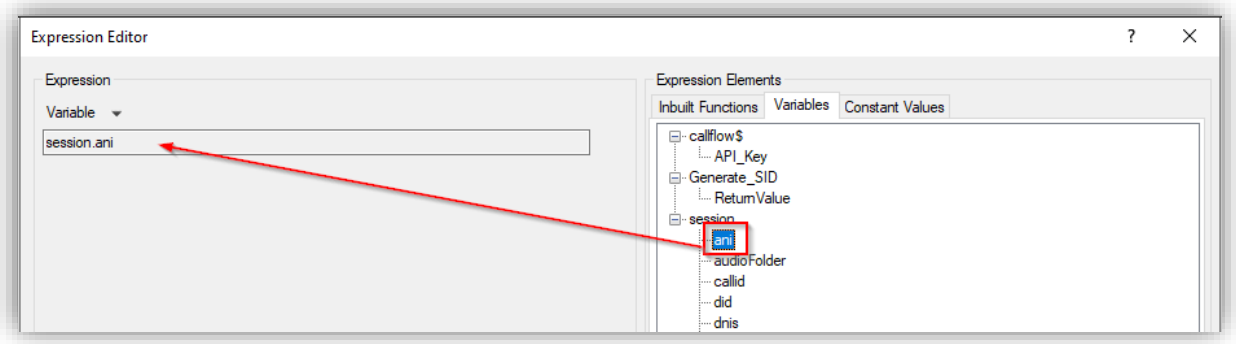
10. In the new window which appears, again, open "**Variables**" tab and drag and drop the "**Return value**" below the "**Generate_SID**" section



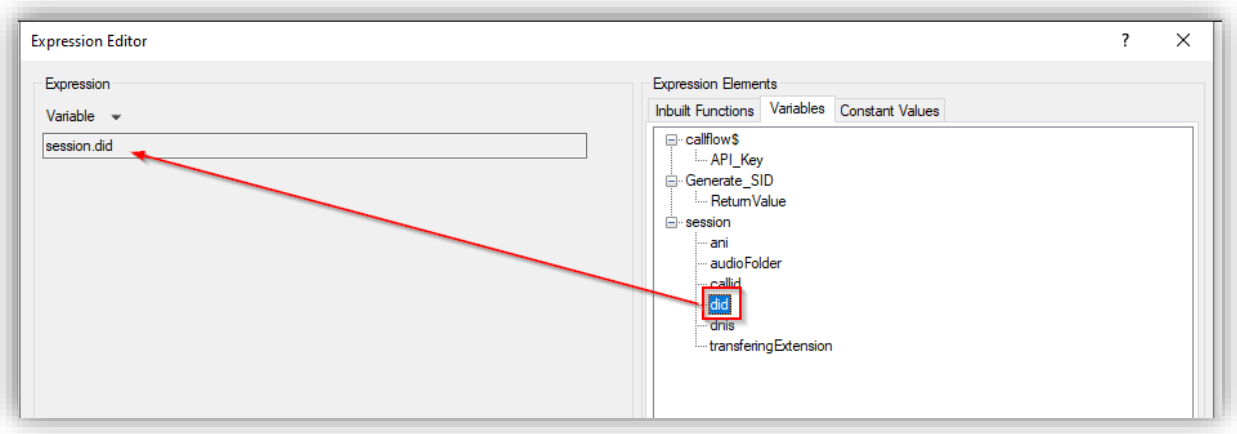
11. Select "Ok". The Input parameters should look like this:



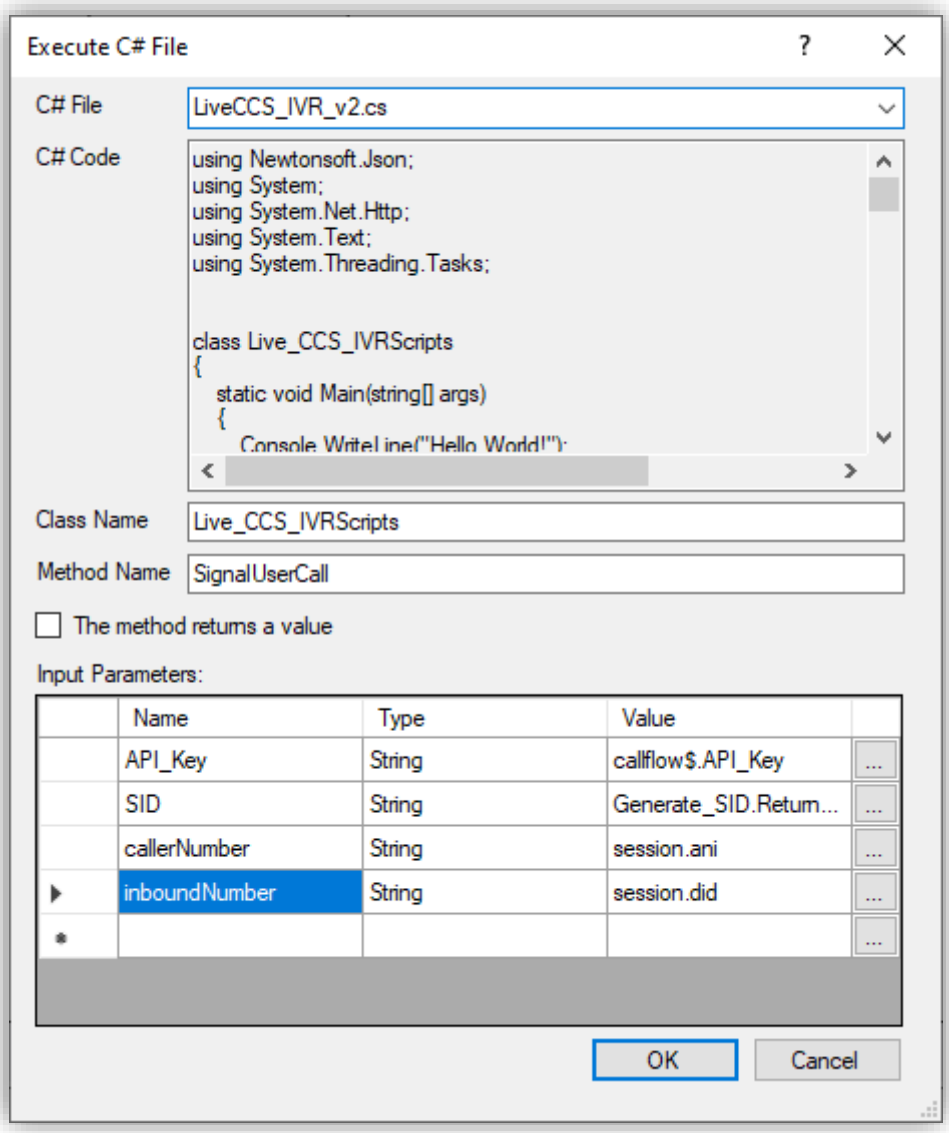
12. Add another parameter named "**callerNumber**", the Type should be "**String**". Click on ellipsis again and select "**ani**" from the "**session**" section. Press Ok.



13. Add another parameter named "**inboundNumber**", the Type should be "**String**". Click on ellipsis again and select "**did**" from the "**session**" section.



14. The whole windows should look like this:

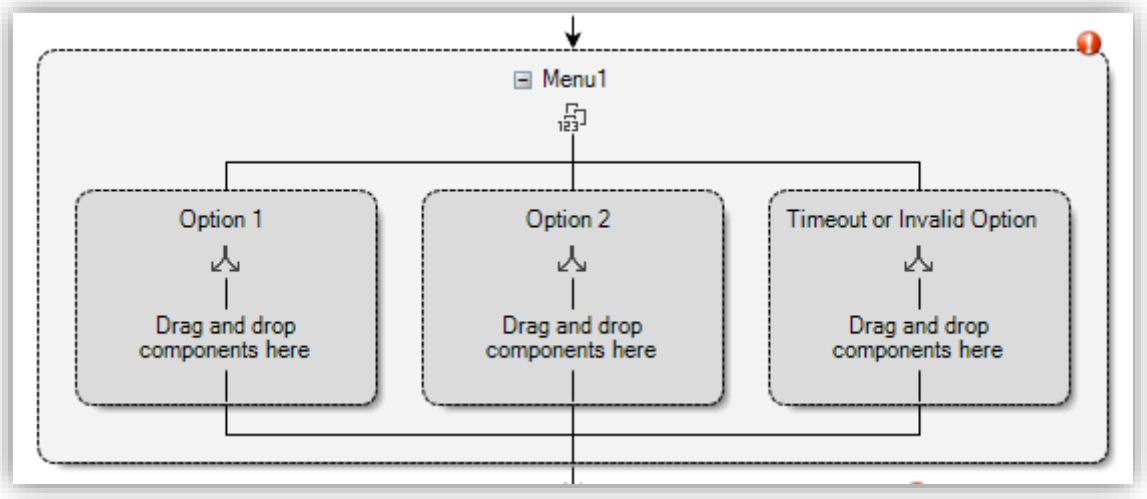


15. Select "**OK**".

Component #3

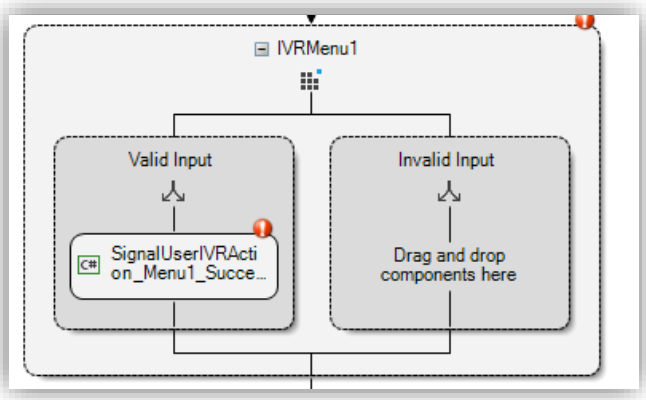
IVR Option select notification

The next step is to signal any user IVR option selection. Apply this procedure after every "**Menu**" block in your flow. We will use an empty menu in this example:



In this example, we will add two **Duplicates** of the same block.

1. Add "**Execute C# File**" block into "**Valid input**" path of your menu and rename it in the properties. Try to add meaningful names, for example "**SignalUserIVRAction_Menu1**"



2. Double click on the newly added block and enter the following information:

C# File: "**LiveCCS_IVR_v2.cs**"

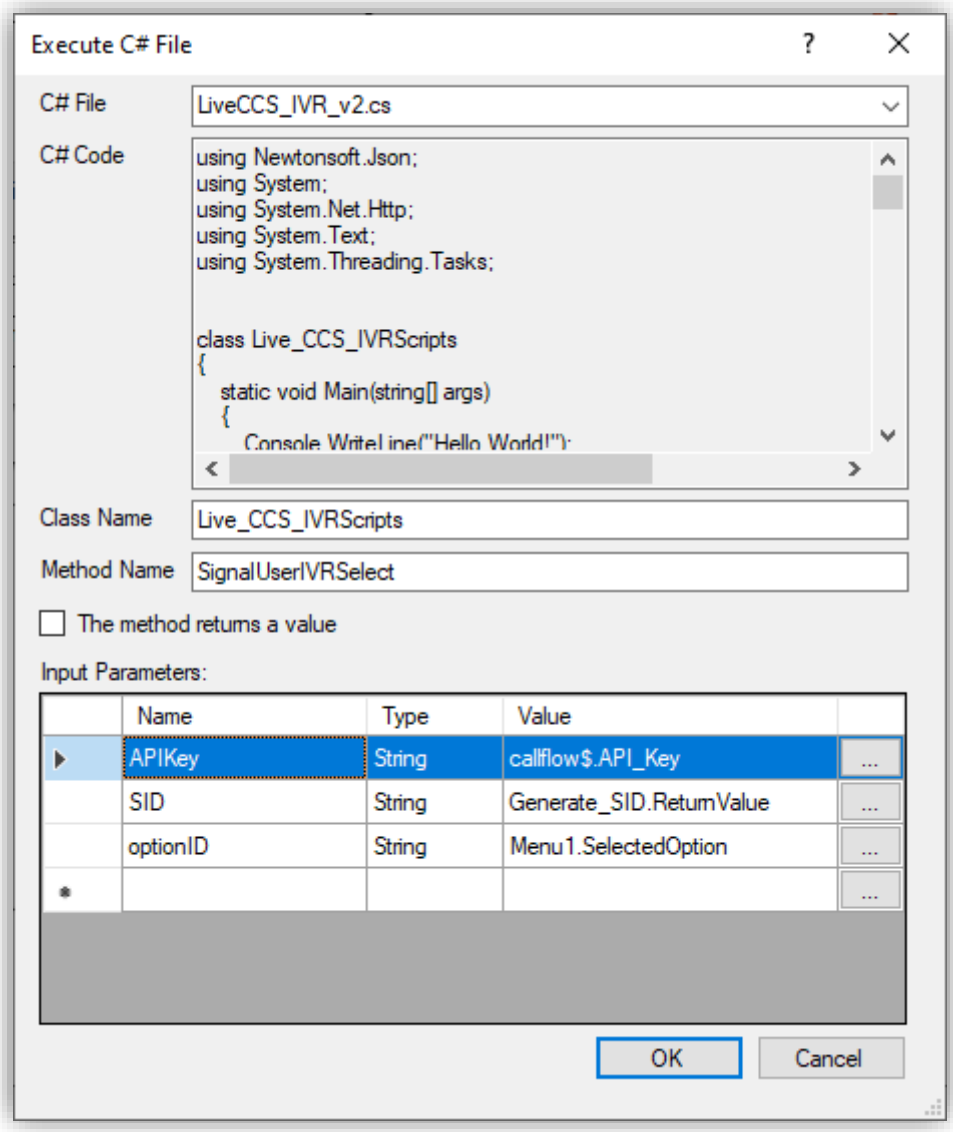
Class Name: "**Live_CCS_IVRScripts"**

Method Name: "**SignalUserIVRSelect**"

This method returns a value: **unchecked**

Input parameters:

Name	Type	Value (Click on Ellipsis)
APIKey	String	Variables -> callflow$ -> API_Key
SID	String	Variables -> Generate_SID -> ReturnValue
optionID	String	Variables -> **<YOUR MENU>** -> **SelectedOption**



3. Press **OK**

Repeat this process for every menu inside your flow.

You must include BOTH VALID AND INVALID Option branches.

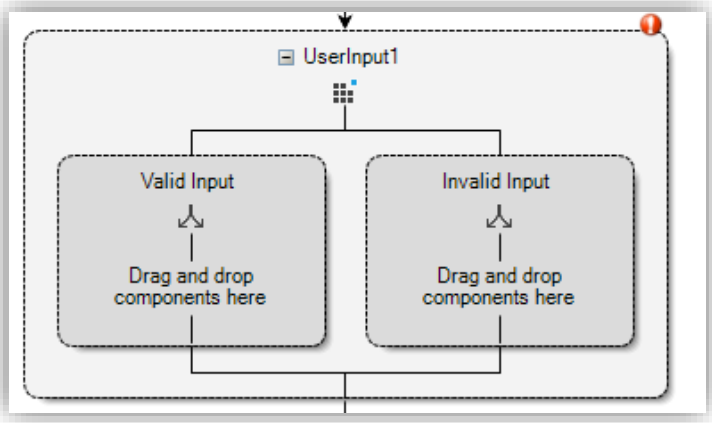
To avoid copying too much information, **you can duplicate your steps**. Click on the step you just created and press Ctrl + C on your keyboard. Then press on the branch you have not yet fixed and press Ctrl + P on your keyboard.

Component #4

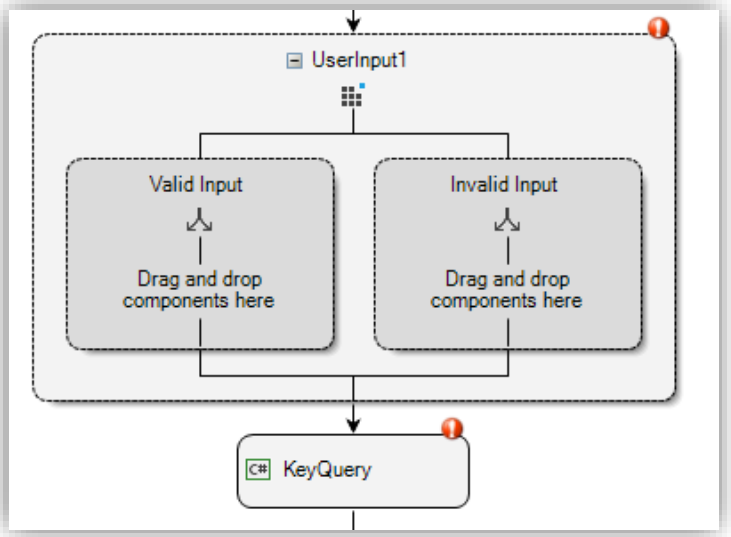
Key Query

Key Query is the last and most important step, where the IVR Flow asks the system whether the input Order number or other information exists and forwards the call to agents.

Apply this step after the "User Input" blocks where you are about to redirect the user.



1. Add "**Execute C# File**" after the user input and rename it to "**KeyQuery**".



2. Double click on the "**KeyQuery**" block and enter the following information:

   C# File: "**LiveCCS_IVR_v2.cs"**

   Class Name: "**Live_CCS_IVRScripts**"

   Method Name: "**KeyQuery**"

   This method returns a value: **checked**

   Input parameters:

Name	Type	Value (Click on Ellipsis)
APIKey	String	Variables -> callflow$ -> **API_Key**
SID	String	Variables -> Generate_SID -> **ReturnValue**
Key1	String	Variables -> **<your user Input>** -> **Buffer**
callerNumber	String	Variables -> session -> **ani**