

Hooks

Hook Workflow Example

This example combines a Hook with the Web Agent's Dynamic Actions Brick to create a simplistic workflow that guides the Agent through ticket creation.

- 1 Create an IVR flow
- 2 Create a Hook
- 3 Set up your Skill
- 4 Set up the Dynamic Actions Brick

Configuration

Create an IVR flow

Name it "WorkflowIVR".

Add a Caller Node pointing to a single Lua Script Node containing the following code.

Save and apply the IVR flow.

Hook Workflow Example IVR flow Lua script

```
local daTag = 'DynamicActions'
local wfParam = 'workflowStep'

local workflowSwitch =
{
  -- Call Reason
  [1] = function()
    setTag(caller, daTag, '{\layout\":{"version\":1,\type\":0,\
settings\":{"selectionMode\":"single\"},"entries\":[{\data\":{"\
title\":"<h3>Call Reason <font color=\\\"#A1D490\\\">Technical Issue<\
/font></h3>\"},\actions\":[{\type\":0,\params\":[{\name\":"id\","
value\":"tcs-samwin-set-tag\"},{\name\":"settings\","value\":{"\
tagKey\":"Reason\","tagValue\":"Technical Issue\"}]}],{\type\":0,\
params\":[{\name\":"id\","value\":"tcs-samwin-trigger-hook\"},{\
name\":"settings\","value\":{"triggerHookForCall\":"true\","
eventName\":"DemoWorkflow\","parameters\":[{\key\":"workflowStep\","
value\":"2\"}]}]}]}],{\data\":{"title\":"<h3>Call Reason <font
color=\\\"#C390D4\\\">Question regarding Invoice</font></h3>\"},\
actions\":[{\type\":0,\params\":[{\name\":"id\","value\":"tcs-
samwin-set-tag\"},{\name\":"settings\","value\":{"tagKey\":"
Reason\","tagValue\":"Invoice\"}]}],{\type\":0,\params\":[{\name\":"
id\","value\":"tcs-samwin-trigger-hook\"},{\name\":"settings\","
value\":{"triggerHookForCall\":"true\","eventName\":"DemoWorkflow\","
parameters\":[{\key\":"workflowStep\","value\":"2\"}]}]}]}],{\
data\":{"title\":"<h3>Call Reason <font color=\\\"#D4A190\\\">Wrong
Number</font></h3>\"},\actions\":[{\type\":0,\params\":[{\name\":"
id\","value\":"tcs-samwin-set-tag\"},{\name\":"settings\","
value\":{"tagKey\":"Reason\","tagValue\":"Wrong Number\"}]}],{\
type\":0,\params\":[{\name\":"id\","value\":"tcs-samwin-trigger-
hook\"},{\name\":"settings\","value\":{"triggerHookForCall\":"
true\","eventName\":"DemoWorkflow\","parameters\":[{\key\":"
workflowStep\","value\":"5\"}]}]}]}]')
  end,
```

```

-- Caller Gender
[2] = function()
    setTag(caller, 'MEMO', '[Add call notes here]')
    setTag(caller, daTag, '{"layout":{"version":1,"type":0,"
settings":{"selectionMode":"single"},"entries":[{"data":{"
title":"<h3>Tag Caller Gender <font color=\\\\"#3399ff\\\\">Male<\/
font><\/h3>"},"actions":[{"type":0,"params":[{"name":"id","
value":"tcs-samwin-set-tag"},"settings","value":{"
tagKey":"CallerGender","tagValue":"Male"}]}],"type":0,"
params":[{"name":"id","value":"tcs-samwin-trigger-hook"},"
name":"settings","value":{"triggerHookForCall":"true","
eventName":"DemoWorkflow"},"parameters":[{"key":"workflowStep",
"value":"3"}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}'}'
end,
-- Wrap Up
[3] = function()
    setTag(caller, daTag, '{"layout":{"version":1,"type":0,"
settings":{"selectionMode":"single"},"entries":[{"data":{"
title":"<h1><font color=\\\\"#64C957\\\\">Click to CREATE TICKET<\/
font><\/h1>"},"actions":[{"type":0,"params":[{"name":"id","
value":"tcs-samwin-trigger-hook"},"settings","value":{"
triggerHookForCall":"true","eventName":"DemoWorkflow"},"
parameters":[{"key":"workflowStep","value":"4"}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}'}'
end,
-- Create Ticket
[4] = function()
    print('Performing REST call to create ticket...')
    setTag(caller, 'Ticket', '1337')
end,
-- Wrong Number / Invalid
[5] = function()
    setTag(caller, daTag, '{"layout":{"version":1,"type":0,"
settings":{"selectionMode":"single"},"entries":[{"data":{"
title":"<h3><font color=\\\\"#68B7E8\\\\">Restart Workflow<\/font><\/
h3>"},"actions":[{"type":0,"params":[{"name":"id","value":
"tcs-samwin-trigger-hook"},"settings","value":{"
triggerHookForCall":"true","eventName":"DemoWorkflow"},"
parameters":[{"key":"workflowStep","value":"1"}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}]}'}'
end
}

local step = tonumber(getParameter(wfParam))
local stepFunction = workflowSwitch[step]
if (stepFunction) then
    print('Triggering Workflow Step ' .. step)

```

```

stepFunction()
else
  print('Invalid Workflow Step ' .. step)
  workflowSwitch[5]()
end

```



Create a Hook

Name it "WorkflowHook".

Add an Event Type Equals condition for "A custom event is triggered".

Add a Custom Event Name Equals condition with name "DemoWorkflow".

Add a Start Script action with "WorkflowIVR".

Apply the Hook configuration.

Information

Enabled

The configured actions will be executed when an event is triggered and all of the conditions are fulfilled.

Conditions

Event type is equal to one of

- A custom event is triggered





Custom event name is equal to one of

- DemoWorkflow





Actions

Start script  WorkflowIVR

Hook

- Apply Configuration
-  Discard Changes
-  Duplicate Hook
-  Delete Hook
-  Rename Hook

Add New Conditions

-  Event Type Equals
-  Queue Equals
-  User Equals
-  Custom Event Name Equals

Set up your Skill

In the Skill that you will be calling into, go to the Call tags tab.

Add a call tag with name "DynamicActions" and the following value.

Apply the Skill configuration.

Hook Workflow Example Skill Call Tag

```

{"layout": {"version": 1, "type": 0, "settings": {"selectionMode": "single"}}, "
entries": [{"data": {"title": "<h3><font color=\"#68b7e8\">Start Workflow<
/font></h3>"}, "actions": [{"type": 0, "params": [{"name": "id", "value": "tcs-
samwin-trigger-hook"}, {"name": "settings", "value": {"triggerHookForCall": "
true", "eventName": "DemoWorkflow", "parameters": [{"key": "workflowStep", "
value": "1"}]}}]}]}]}]}]}

```

Call tags

- 🔧 Change Settings
- 🖨️ Print Page
- 📝 Custom Note

Call tags

Call tags set when call is assigned

- The call tag 'DynamicActions' will be set to `'{"layout":{"version":1,"type":0,"settings":{"selectionMode":"single"},"entries":[{"data":{"title":"<h3> Start Workflow</h3>"},"actions":[{"type":0,"params":{"name":"id","value":"tcs-samwin-trigger-hook"}, {"name":"settings","value":{"triggerHookForCall":"true","eventName":"DemoWorkflow","parameters":{"key":"workflowStep","value":"1"}}}}]}}'`

Set up the Dynamic Actions Brick

In the Web Agent, add a `Dynamic Actions Brick` to your brick wall.

Enter its settings and configure it to use call tag "DynamicActions".

Apply the settings.

Brick Configuration: Dynamic Actions

Title:	Dynamic Actions	✕
Scaling:	100%	▼
Configuration:	<input checked="" type="checkbox"/> Use a Call Tag	
	📄 DynamicActions	✕

Usage

Call into the Skill that you configured the call tag for.

The Dynamic Actions Brick will show a **Start Workflow** button that you can click.