Description

The **Ayehu NG** solution features a robust turn-key **ServiceNow** integration that allows users to interact with every form and field in their **ServiceNow** instance, whether they be standard or custom. This tutorial will show you how to use the out-of-the-box plug-and-play **ServiceNow** activities to dynamically retrieve a list of attachments associated with a specific ticket.

Before following this tutorial, ensure that the "Attachment(sys_attachment)" form and its fields are discovered in your **ServiceNow** module.

For more information on permission requirements for accessing ticket attachments in **ServiceNow**, visit our "*ServiceNow Integration Module Access/Permissions Requirements*" support article: https://support.ayehu.com/hc/en-us/articles/360048555853-ServiceNow-Integration-Module-Access-Permissions-Requirements

For more information on discovering forms and fields in **ServiceNow**, visit our "*Integrating with ServiceNow*" support article and scroll down to the section labeled "Forms": https://support.ayehu.com/hc/en-us/articles/360006651853-Integrating-with-ServiceNow

Solution

In the workflow below, we are storing a **ServiceNow** ticket number and then cross-referencing its "Sys ID" field with the "Attachment" form to retrieve a list of all attachments associated with it.

First, we use the **MemorySet** activity to store a ticket number in the **%ticketNumber%** variable, as seen below. In this example, we're using a ticket on the "Incident(incident)" form.
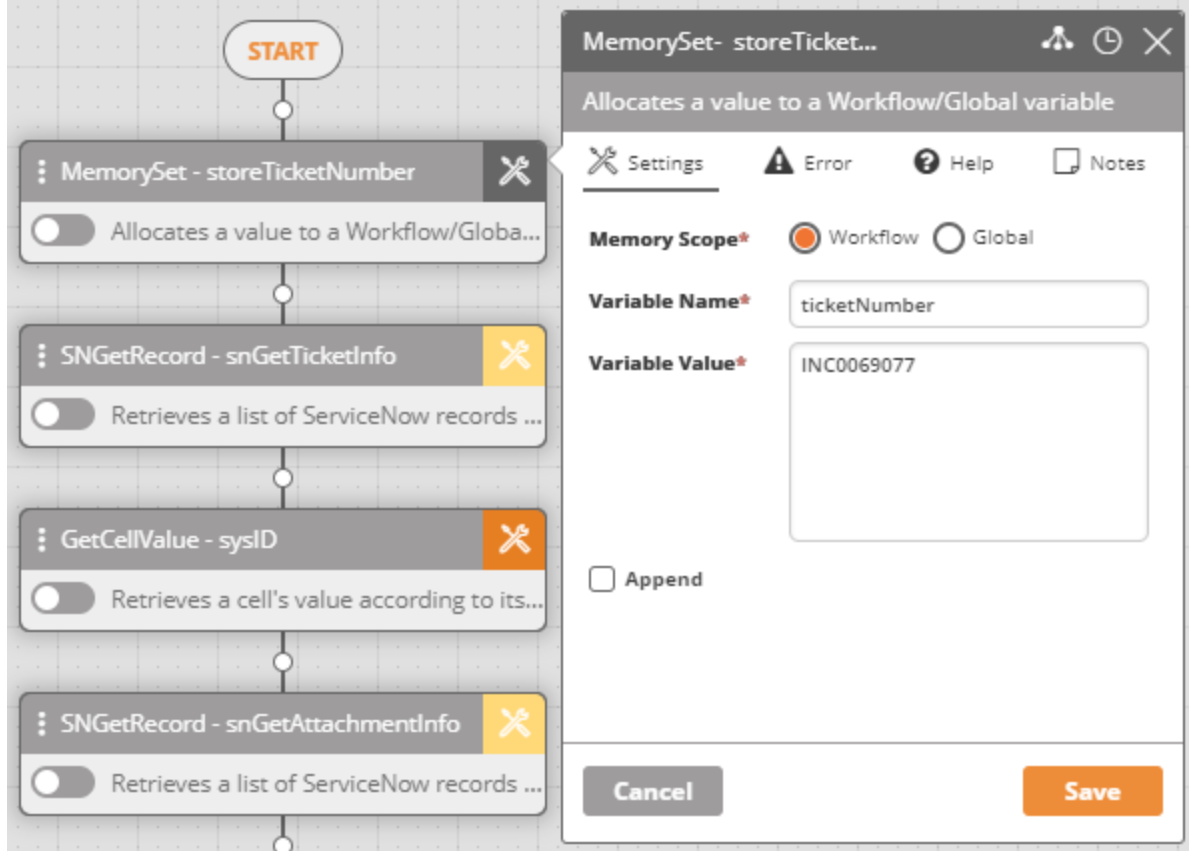
Next, we use the **SNGetRecord** activity to retrieve all of the fields for the ticket number stored in **%ticketNumber%** and return them in a result-set table named **%snGetTicketInfo%**.
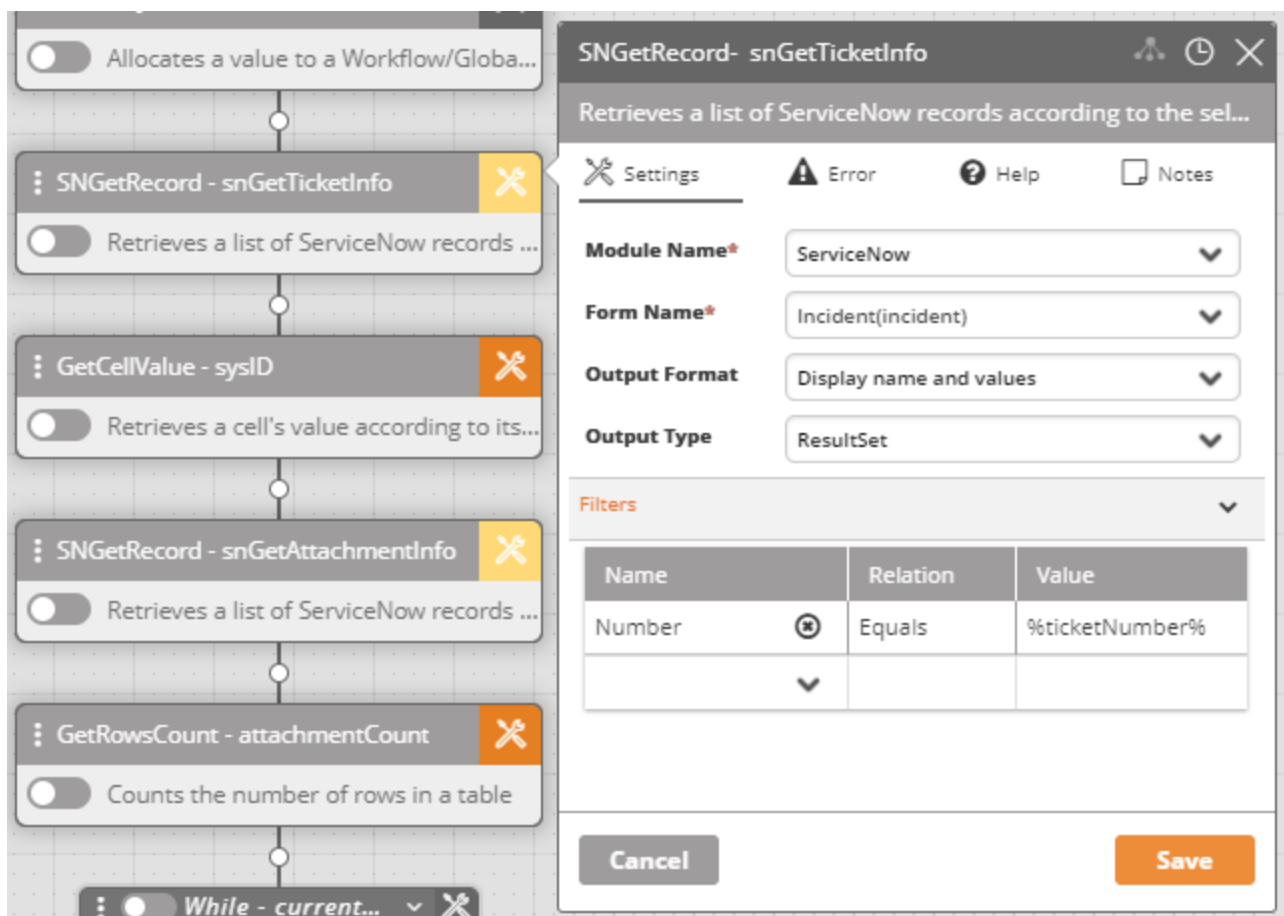
The next screenshot shows us using the **GetCellValue** activity to pull the "Sys ID" field from the **%snGetTicketInfo%** result-set table returned by the **SNGetRecord** activity.

With our new **%sysID%** variable, we again use **SNGetRecord** to return a result-set table of all records on the "Attachment(sys_attachment)" table whose "Table sys ID" field is equal to **%sysID%** (the unique ID of the parent ticket).

Now that we have a result-set table containing all attachments associated with the ticket, we use the **GetRowsCount** activity to store the number of rows contained in that attachment list (stored in the **%snGetAttachmentInfo%** table).
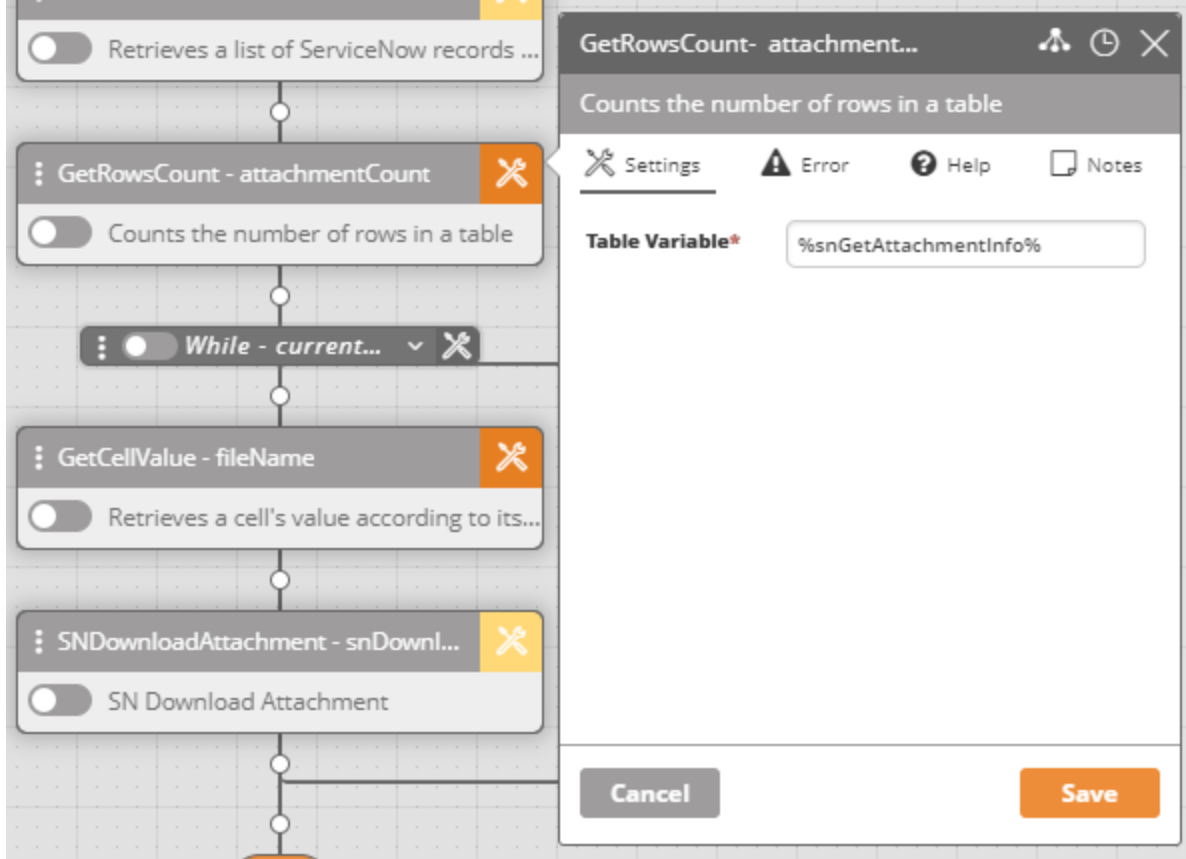
We then create a while-loop, as seen below. In this example, the while-loop has been named **%currentAttachment%** and our counter is set to the output of the previous **GetRowsCount** activity.
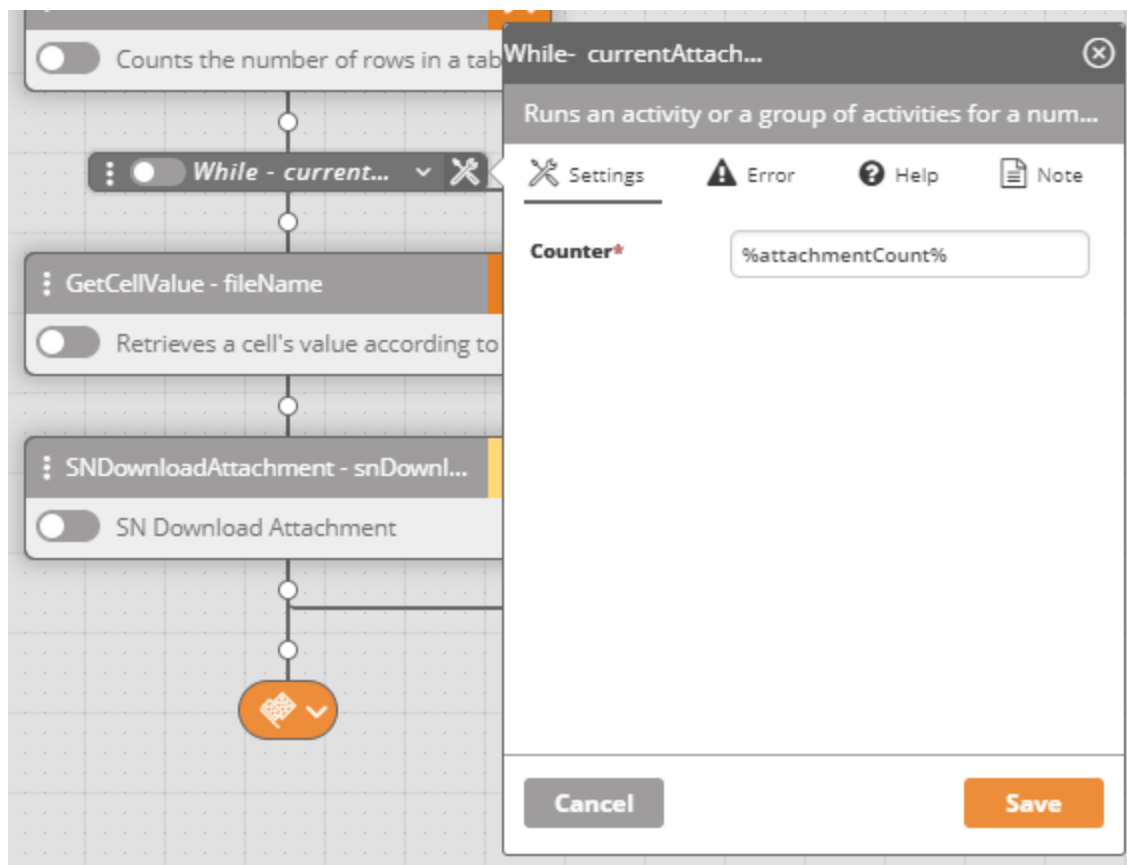
Using **GetCellValue**, we retrieve the "File name" column corresponding to the current row (**%currentAttachment%**) of the **%snGetAttachmentInfo%** result-set table. We have named this **GetCellValue** activity to **%fileName%**.

As seen in the screenshot below, our last step is to use the **SNDownloadAttachment** activity, passing to it the **%ticketNumber%** variable we hard-coded at the beginning of the workflow, as well as the **%fileName%**variable from the previous step. For this example, we're using "C:\temp\" as our output directory.

Successful Workflow Execution

In the **Workflow Execution Log** screenshot below, we can see how the unique "Sys ID" field was retrieved for the **ServiceNow** ticket and then referenced to return a result-set table of all attachments associated with it. We can also see three (3) iterations of the while-loop as we retrieved the filename for each attachment and then downloaded it.

Event Type	Activity Name	Status	Result	
Incoming event				
MemorySet	storeTicketNumber	Executed	Success	
SNGetRecord	snGetTicketInfo	Executed	Result	»
GetCellValue	sysID	Executed	85cc00901b351c1070950dc8cc4bcba3	
SNGetRecord	snGetAttachmentInfo	Executed	Result	»
GetRowsCount	attachmentCount	Executed	3	
GetCellValue	fileName	Executed	ayehu.jpg	
SNDownloadAttachment	snDownloadFileAttachment	Executed	Success	
GetCellValue	fileName	Executed	AyehuLogo_UPDATED.png	
SNDownloadAttachment	snDownloadFileAttachment	Executed	Success	
GetCellValue	fileName	Executed	Tuxterminal_icon.png	
SNDownloadAttachment	snDownloadFileAttachment	Executed	Success	
Terminate		Executed		