

Add and edit ServiceNow transform map scripts for Ivanti

Fluorine+

To ensure that the data being shared out of Ivanti and into ServiceNow (inbound data) is being mapped correctly when arriving at your ServiceNow instance, you will next need to add and update some of the scripts in the **PSP Common Incident to Incident** transform map under the PSP Common Transform Maps module. [ServiceNow transform maps](#) handle the format of data being shared into a ServiceNow instance. Making changes to the transform map scripts will allow data to be properly shared out of Ivanti Service Manager and then be mapped correctly to the relevant fields in ServiceNow.

Prerequisites

- ⚠ First, you will need to [create a ServiceNow dynamic share for Ivanti](#).
- ⚠ You will also need to [create a ServiceNow subscribed queue and subscribe for Ivanti](#) and [edit ServiceNow outbound table map fields for Ivanti](#).

Procedure

To add and edit the PSP Common Incident to Incident transform map scripts for Ivanti, follow these steps:

1

Access the PSP Common Incident to Incident transform map

Log into your ServiceNow instance and navigate to **System Import Sets > Administration > Transform Maps** or simply type and then select **Transform Maps** in the Filter Navigator on the upper left-hand side of the screen. Then, search for and then click into the **PSP Common Incident to Incident** transform map.

2

Edit the onAfter script with Order 110

Scroll down to the tabs at the bottom of the form and click the **Transform Scripts** tab. In the list of **Transform Scripts**, search for and click into the **onAfter** script with an **Order** of **110**. Scroll down to the scripting window and replace "**msp_client_incident_sent**" with "**ivanti_sent**"

Finally, click **Update** in the bottom left-hand corner of the form to save the changes to your transform map script.

3

Add transform scripts

Create **New** transform map scripts according to the information in the table shown below.

When	Order	Script
------	-------	--------

onBefore	80	<pre> // Prevents duplication of comments (function runTransformScript(source, map, log, target /*undefined onStart*/) { if (source.u_comments.nil()) return; var elementId = target.sys_id; if (!source.u_correlation_id.nil()) elementId = source.u_correlation_id; else if (!source.u_sys_id.nil()) { var igr = new GlideRecord('incident'); igr.addQuery('correlation_id', source.u_sys_id); igr.query(); if (igr.next()) { elementId = igr.u_value; } } var sgr = new GlideRecord("sys_journal_field"); sgr.addQuery("name", 'incident'); sgr.addQuery("element_id", elementId); sgr.addQuery("value", source.u_comments); sgr.query(); if (sgr.next()) { ignore = true; } })(source, map, log, target); </pre>
onBefore	85	<pre> (function runTransformScript(source, map, log, target /*undefined onStart*/) { if (source.u_caller_email.nil()) { return; } var gr = new GlideRecord("sys_user"); gr.addQuery("email", source.u_caller_email); gr.query(); if (gr.next()) { target.caller_id = gr.sys_id; } })(source, map, log, target); </pre>
onAfter	90	<pre> (function runTransformScript(source, map, log, target /*undefined onStart*/) { var pspS = new PerspectiumEncryption(); var pspR = new PerspectiumReplicator(); var ogr = new GlideRecord("psp_out_message"); ogr.addQuery("state", "deferred"); var orQc = ogr.addQuery("u_extra", "CONTAINS", "number=" + source.u_number); orQc.addOrCondition("u_attributes", "CONTAINS", target.sys_id); ogr.query(); while (ogr.next()) { // update value with correlation id before we send out var decodeData = pspS.decryptString(ogr.value); decodeData = decodeData.replace("<correlation_id/>", "<correlation_id>" + target. correlation_id + "</correlation_id>"); if (decodeData.indexOf("</attachments>") > -1) { ogr.name = ogr.name.replace("update", "attach"); } ogr.value = pspS.encryptStringWithCipher(decodeData, "3"); ogr.u_extra = ""; ogr.state = "ready"; ogr.update(); } })(source, map, log, target); </pre>

onAfter	95	<pre> (function runTransformScript(source, map, log, target /*undefined onStart*/) { if (!source.u_correlation_id.isNil()) { return; } var pspR = new PerspectiumReplicator(); var ds = new GlideRecord("psp_replicate_conf"); ds.addQuery("u_target_queue.u_name", "psp.in.meshlet.ivanti"); ds.addQuery("active", "true"); ds.addQuery("sync_direction", "share"); ds.query(); if (!ds.next()) { return; } var gr = new GlideRecord("incident"); gr.addQuery("sys_id", target.sys_id); gr.query(); if (gr.next()) { pspR.shareRecord(gr, "incident", "update", ds.sys_id); } })(source, map, log, target); </pre>
---------	----	--

Next steps

[Add ServiceNow attachment.uploaded script action for Ivanti](#)

Similar topics

- [Create Ivanti business rules](#)
- [Create a custom Correlation ID field for Ivanti](#)
- [Get started with ServiceBond for Ivanti](#)
- [Bond Ivanti with another app](#)
- [Create a ServiceNow subscribed queue and subscribe for Ivanti](#)

Contact Perspectium Support



US: 1 888 620 8880

UK: 44 208 068 5953

support@perspectium.com