ONE IDENTITY™

One Identity Manager 8.0.2

Administration Guide for Connecting to ServiceNow

One Identity Manager Administration Guide for Connecting to ServiceNow
Updated - August 2018
Version - 8.0.2

Contents

Managing ServiceNow

One Identity Manager offers simplified integration with ServiceNow, which allows users of both systems to navigate easily from ServiceNow into One Identity Manager and gives users a "one stop shop" to request all IT related items.

One form of integration between One Identity Manager and ServiceNow provides ServiceNow the control to act as a master.

The integration between ServiceNow and One Identity Manager, provides mutual customers a complementary identity access governance and service management solution. This solution works to ensure that clear audit trails and strong controls are in place to meet ever stringent security and compliance requirements around user access to sensitive applications. One Identity Manager does this by creating service request tickets within ServiceNow, when a request for access is submitted from One Identity Manager and the fulfillment of the request requires manual completion by an IT service agent.

For example, when an end-user requests application access (SAP, etc) in One Identity Manager and that request requires manual fulfillment (no automated provisioning is available / preferred), the integration creates a ticket in ServiceNow. One Identity Manager then tracks the request by polling ServiceNow for the ticket's status periodically. As the IT service agent updates the status of the ticket the changes are reflected in One Identity Manager. Once a ticket is closed or complete, One Identity Manager records this and closes the request. This is all tracked within One Identity Manager so that it can be reported on and provided as part of an audit.

Another form of integration between One Identity Manager and ServiceNow provides the control to One Identity Manager to act as the master. Tickets are created for service items that have their service categories enabled for ServiceNow. These requests are subjected to the regular approval policies of the Identity Manager and the changes done on the requests are updated to ServiceNow allowing users to check the status of the requests or their tickets on ServiceNow.

Technical Overview

To access ServiceNow data, the ServiceNow Component is installed on a synchronization server. The ServiceNow Component establishes communication with the ServiceNow Cloud through the ServiceNow REST APIs. This ensures that tasks such as creating and tracking tickets can be configured through the One Identity Manager interface.

Figure 1: One Identity Manager and ServiceNow Integration

**One Identity
Manager
Database**

**One Identity Manager
Job Server**
ServiceNow Component
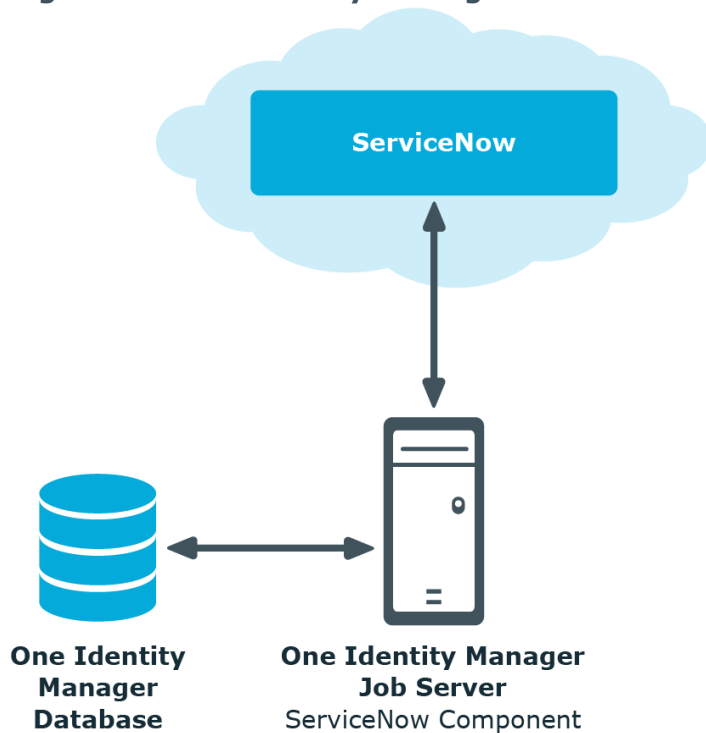
The integration solution uses the following process chains to create and track service tickets:

- SCN_Create ServiceNow ticket: This process chain uses the **CreateTicket** SCN process task.

- SCN_Get ServiceNow ticket status: This process chain uses the **GetTicketStatus** SCN process task.

- SCN_Update One Identity Manager status to ServiceNow: This process chain uses the **UpdateServiceNowStatus** SCN process task.

2

Installation of ServiceNow module

ServiceNow module is similar to other One Identity Manager modules and follows the same installation model. For information on installation of ServiceNow module, refer the **Installing One Identity Manager Components** section of the *One Identity Manager Installation Guide*.

Configuring ServiceNow module

Before proceeding with the ServiceNow configuration, ensure that a job server is assigned to the **SCN-Connector-Ext** role. Configuring ServiceNow module involves the following:

- Configuring ServiceNow Connection Parameters
- Enabling service categories for ServiceNow
- Configuring One Identity Manager Web Portal Authentication Modules

Configuring ServiceNow Connection Parameters

To support the One Identity Manager and ServiceNow integration solution, the following ServiceNow connection parameters must be assigned in the One Identity Manager Designer.

- InstanceURL
- InstanceUser
- InstanceUserPassword

To configure the ServiceNow connection parameters

1. In One Identity Manager Designer, navigate to **Base Data** | **General** | **Configuration parameters**.

2. In the Configuration parameters pane, expand **SCN**.

   The SCN configuration parameters are displayed.

3. To assign the parameters to the Designer:

   a. Click **InstanceURL**, and in the Configuration parameters dialog, in the **Value** field enter the instance url for ServiceNow.

b. Click **InstanceUser**, and in the Configuration parameters dialog, in the **Value** field enter the instance user for ServiceNow.

c. Click **InstanceUserPassword**, and in the Configuration parameters dialog, in the **Value** field enter the instance user password for ServiceNow.

4. On the Toolbar, click **Commit to database**. Follow the on-screen instructions displayed to save the changes to the database.

5. On the Menu bar, click **Database | Compile database**. Follow the on-screen instructions displayed in the Designer - Database Compiler wizard to compile the database.

The ServiceNow connection configuration parameters are now assigned in the One Identity Manager Designer.

Enabling service categories for ServiceNow

To enable service categories for ServiceNow

1. Navigate to **Manager | IT Shop | Service Catalog**.

The Service categories pane displays the available service categories.

2. Right-click the service category for which you want to enable ServiceNow, and select **Tasks | Change master data**.

3. On the service category dialog, select the **Enable for ServiceNow** check box.

4. On the toolbar, click **Save**.

The selected service category is now enabled for ServiceNow. Repeat the steps to enable other available service categories for ServiceNow.

Configuring One Identity Manager Custom IT Shop for ServiceNow module

The ServiceNow module requires the creation of a customized approval policy, which must be bound to the IT shop elements. This process involves the following:

1. Creating approval policies for external approval.

2. Assigning external approval policy to IT Shop.

Creating approval policies for external approval

Before working with the One Identity Manager IT shop, you must configure the approval policies to require approval from external sources.

To create Approval polices

1. Navigate to the **Manager | IT Shop | Basic Configuration Data** . Right-click **Approval workflows**, and select **New**.

1. On the Approval workflow pane, go to Toolbox, under **Workflow**, click **Edit**.

2. In the Edit workflow dialog, in the **Name** field, enter a name for the new workflow and click **Ok**.

   A new workflow with the assigned name is displayed in the workflow pane.

3. To add a new approval level to the approval workflow, in the **Toolbox**, under **Approval levels**, click **Add**.

   A dialog to edit the selected approval step properties is displayed.

4. Enter the values in the following fields:

   - **Single step**: Enter an appropriate name.
   - **Approval procedure**: From the drop-down menu, select **EX-Approvals to be made externally**.
   - **Processing Status on success**: From the drop-down menu, select **Assigned by self-service**.
   - **Event**: Add an appropriate name.

5. Enter values in other required fields and click **Ok**.

6. On the Toolbar, click **Save** to save the approval workflow.

7. To create a new approval policy, navigate to **Manager | IT Shop | Basic Configuration Data**. Right-click **Approval policies**, and select **New**.

8. In the Approval policy pane, enter the values in the following fields:

   - **Approval policy**: Enter an appropriate name for the policy
   - **Approval workflow**: From the drop-down select the newly created approval workflow.

9. Click **Save**.

Assigning external approval policy to IT Shop

To assign a new external approval policy to IT shop

1. Create a new IT Shop or a new shelf to an existing IT Shop.

2. Modify the approval policy for the IT Shop with the newly created approval policy.

For more information on assigning external approval policy to IT shop, see the *One Identity Manager Administration Guide*.

Using ServiceNow as a One-Stop Shop

After the integration of One Identity Manager IT shop into the ServiceNow portal, ServiceNow web portal serves as a one-stop shop for requesting ServiceNow as well as One Identity Manager items.

The integration is accomplished by combining the External Content View of a content item in ServiceNow and the deep linking feature of the One Identity Manager Web portal page.

For ServiceNow web portal to serve as a one-stop shop you must configure the ServiceNow web portal and One Identity Manger web portal authentication modules.

ⓘ NOTE: Before configuring the One Identity Manager web portal, the ServiceNow portal must be configured to display the external content item.

Configuring ServiceNow web portal

To configure the ServiceNow portal

1. Login to the ServiceNow web portal.

2. Create a new category:

    a. On the ServiceNow home page, on the Navigation bar, select **Service Catalog | Catalog Definitions | Maintain Categories.**

    b. The Categories list is displayed.

    c. On the Category pane, click **New**.

    d. On the Category **New Record** pane, in the **Title** field, enter the required category title.

    e. In the **Catalog** field, click the **Search** icon to search for a catalog in the catalog list and assign it.

    f. Click **Submit**.

3. Create a new content Item:

   a. On the Navigation bar, select **Service Catalog | Catalog Definitions | Content Items**.

   b. On the **Content Items** pane, click **New**.

   c. On the **Content Item New Record** pane, from the **Content Type** drop-down menu, select **External Content**.

   d. In the **URL** field, paste the url **https://<WebPortalServer>/IdentityManager/page.axd?ContextID= <PageLink>** for deep links to One Identity Manager.

      For example, **https://webportalserver.com/IdentityManager/page.axd?ContextID =VI_ITShop_ProductSelection**

      The context ID changes based on the **Web Designer** page that must be displayed.

4. Assign a catalog and category to the newly created content item.

   a. On the Content Items pane, click 🔒 against **Catalogs** and then click 🔍 to search and select a catalog.

   b. Click 🔍 against **Category** to search and select a category from the assigned catalog.

   c. Click **Update**.

   The catalog and category are assigned to the newly created content item.

5. Add the newly created category to the **Service Catalog** page.

   a. Navigate to **Self Service | Service Catalog**.

   b. Click the + button on the upper right corner of the screen. Search for the newly created category and click **Add here** to add the category on to the screen in the required place.

Configuring One Identity Manager Web Portal Authentication Modules

To configure authentication modules on the One Identity Manager web portal

1. Go to the One Identity Manager Installers location, navigate to Modules | TST | Assemblies. and run the **WebDesigner.ConfigFileEditor** file.

2. On the file browser, navigate to the One Identity Manager web portal root folder, select the **web.config** file, and click **Open**.

3. The **Web Designer Configuration Editor** page is displayed.

4. On the Web Project pane, in the Authentication module drop-down menu select the required authentication module. For example, Employee (Role based), OAuth 2.0 / OpenID Connect (role based), and so on.

ⓘ NOTE: The login behavior of One Identity Manager may vary when the link is accessed from the ServiceNow web portal, based on the web portal authentication module.

One Identity Manager login behaviour in ServiceNow portal

The login behavior of One Identity Manager web portal for **Employee (Role based)** authentication module is described below:

- When you click the One Identity Manager link in ServiceNow web portal for the first time, an authorization page to the One Identity Manager web portal is displayed. You are prompted to enter the user credentials of One Identity Manager to log in to the system.

- After logging in, the One Identity Manager web portal displays the available categories.

- You can log in with a different ServiceNow user to another instance of ServiceNow portal, in the same browser using a different tab. In this case, when you click the One Identity Manager link in ServiceNow web portal, the One Identity Manager web portal is displayed by default without requiring you to enter the login credentials. One Identity Manager uses the credentials of the last logged in user.

- When the browser is closed and a new instance of the browser is opened, clicking the One Identity manager link in the ServiceNow web portal displays the authorization page to the One Identity Manager web portal prompting you to enter the user credentials.

The login behavior of One Identity Manager web portal for **OAuth 2.0 / OpenID Connect (role based)** authentication module is described below:

- When you click the One Identity Manager link in ServiceNow web portal for the first time, an authorization page to the One Identity Manager web portal is displayed prompting you to enter the user credentials of One Identity Manager.

- After logging in, the One Identity Manager web portal displays the available categories.

- You can log in with a different ServiceNow user to another instance of ServiceNow portal, in the same browser using a different tab. In this case, when you click the One Identity Manager link in ServiceNow web portal, the One Identity Manager web portal is displayed by default without requiring you to enter the login credentials. One Identity Manager uses the credentials of the last logged in user.

- When the browser is closed and a new instance of the browser is opened, clicking the One Identity manager link in the ServiceNow web portal displays the One Identity Manager web portal without requiring you to enter the login credentials. One Identity Manager uses the credentials of the last logged in user.

Logging out from ServiceNow

When a user logs out of the ServiceNow portal, the process to log off from the One Identity Manager Web session starts. ServiceNow allows customization of the log off from the portal, which can be used to browse the One Identity Manager deep link which clears the authentication session.

One Identity Manager 8.X does not have a log out URL available for clearing the SSO session, which must be implemented in One Identity Manager OAuthenticator module to solve this use-case. Feature ID aimed at solving the impediment is 771671. A workaround for the log out issue is to log off the One Identity Manager session, before logging out of the ServiceNow portal.

Use Case Scenarios

This section provides descriptions about the following use case scenarios:

- Use Case Scenario with ServiceNow as Master
- Use Case Scenario with One Identity Manager as Master

Use Case Scenario with ServiceNow as Master

In this use case scenario, a ticket is created on ServiceNow for any item that is requested on the One Identity Manager web portal with ServiceNow as the master.

This scenario uses the following steps:

Step 1: **Enable Service category**: To enable the functionality of creating a ServiceNow ticket for a service item, you must first enable the service category for ServiceNow.

Step 2: **Create service request ticket**: The process chain **SCN_Create ServiceNow ticket** creates a ticket on ServiceNow, based on the parameters configured in the process task. After a ticket is created, it must be resolved on ServiceNow, as ServiceNow is the master.

Step 3: **Check ticket status**: The process chain **SCN_Check_status_of_the_ ServiceNow_ticket** is run on a scheduled basis to check for the **PersonWantsOrg** entries that are still open and associated with a ticket in ServiceNow.

The process chain returns **SNOWResponse** variable, which contains the status of the ticket from ServiceNow. This response is used to change the associated **PersonWantsOrg** entry in One Identity Manager.

> ⓘ NOTE: The schedule for checking the status of the ticket can be set, by changing the schedule in the **Designer | Base Data | General | Schedules**. Select the schedule **Service now Ticket status check schedule**.

In an error scenario where the ticket is not created for a PWO request entry, the request goes into the denied state automatically and the user can request for the same product again.

Use Case Scenario with One Identity Manager as Master

In this use case scenario, a ticket is created on ServiceNow for any item that is requested on the One Identity Manager web portal with One Identity Manager as the master.

This scenario uses the following steps:

Step 1: **Enable Service category**: To enable the functionality of creating a ServiceNow ticket for a service item, you must first enable the service category for ServiceNow.

Step 2: **Create service request ticket**: The process chain **SCN_Create ServiceNow ticket** creates a ticket on ServiceNow, based on the parameters configured in the process task. After a ticket is created, it must be resolved on ServiceNow, as ServiceNow is the master.

Step 3: **Check ticket status**: Once a ticket is created, it must go through the regular approval processes that is configured to the shelf containing the item. The process chain **SCN_Update 1IM status to ServiceNow** is triggered if any change is made on the status of the **PersonWantsOrg** entry. This process chain ensures that the JSON response is sent to the ServiceNow end, and the associated ticket is updated in ServiceNow.

> ❶ NOTE: If the process chain **SCN_Update 1IM status to ServiceNow** fails, it is recommended to create a new **MailComponent** step that enables you to send a mail alert to the requester.

In case of an error, when the ticket is not created for a PWO request entry, the request goes into denied state and then, the user is allowed to request for the same product again.

Customizing ServiceNow module

Customizing the ServiceNow module involves the following steps:

- Creating ServiceNow Ticket – Adding more attributes to the ticket creation
- Manipulating the response from ServiceNow
- Constructing the Request for updating One Identity Manager status to ServiceNow ticket

Creating ServiceNow Ticket – Adding more attributes to the ticket creation

The ServiceNow component **CreateTicket** enables you to create tickets.

To create tickets

1. In the One Identity Manager Designer, navigate to **Process Orchestration | Process Components | SCNComponent | CreateTicket**.

2. To create a parameter for the CreateTicket task, right-click **CreateTicket**, and select **New parameter**.

   The Parameters properties dialog is displayed.

   a. In the **Name** field, enter a name that matches the ServiceNow parameter name. This new parameter must match the ServiceNow parameter. For example, if you want to add the ServiceNow parameter **resolved_by**, provide the name for the new One Identity Manager parameter as **resolved_by**.

   b. In the **Value template** field, enter the value template with the syntax: **Value = <*Value to be configured>*. For example, **Value = "test"**.

A process step with the **CreateTicket** task sends the configured values to ServiceNow to create a ticket. The process chain **Create Service Now Ticket** uses this process step to create a ticket.

Manipulating the response from ServiceNow

The process step with **GetTicketStatus** task gets the status of a ticket.

The process chain **SCN_Create ServiceNow Ticket** enables to create a ticket in ServiceNow and update the field **ServiceNowSystemID** of the **PersonWantsOrg** table. This field is used in the **SCN_Check_status_of_the_ServiceNow_ticket** process chain later, which is called over time. This process chain contains a process step using the **GetTicketStatus** task which uses the **ServiceNowSystemID** as a reference to the ServiceNow and returns the status in the field SNOWResponse.

SNOWResponse value is a JSON response which is then parsed to get any desired field of the ServiceNow response schema. These fields can later be leveraged to change the status of the associated **PersonWantsOrg** entry accordingly.

This SNOWResponse value is used in the script **SCN_UpdatingOneIMTicketStatus**. This is first parsed to a NewtonSoft JObject type and then values are taken out using the SelectToken function. Further explanation is given below.

The response value script is represented below:

```
#If Not SCRIPTDEBUGGER

References Newtonsoft.Json.dll

#End If

'This script is used for updating the status from the ServiceNow ticket to the
corresponding PersonWantsOrg entry in OneIM.

'Dieses Skript wird zum Aktualisieren des Status vom ServiceNow-Ticket zum
entsprechenden PersonWantsOrg-Eintrag in OneIM verwendet.

Public Overridable Function SCN_UpdatingOneIMTicketStatus( ByVal statusResultsData As
String , ByVal UID_PWO As String) As Boolean

Dim PWO As ISingleDbObject = Connection.CreateSingle("PersonWantsOrg", UID_PWO)

Dim snowResponse As Newtonsoft.Json.Linq.JObject = Nothing

Try

snowResponse = Newtonsoft.Json.Linq.JObject.Parse(statusResultsData)

'We can retrieve any value from the ServiceNow Response JSON.

'In this script, we are getting the values for "sys_id", "number", "state", etc.

'A detailed sample of a JSON response can be found in the OneIM Administration Guide.

'Wir können jeden Wert aus dem ServiceNow-Antwort-JSON abrufen.

'In diesem Skript erhalten wir die Werte für "sys_id", "number", "state" usw.
```

```vbnet
'Ein detailliertes Beispiel einer JSON-Antwort finden Sie im OneIM-
Administrationshandbuch.

'Syntax to get a value from the json : snowResponse.SelectToken("<variable
name>").ToString()

'Syntax, um einen Wert aus dem json: snowResponse.SelectToken ("<Variablenname>") .
ToString ()


'Example : snowResponse.SelectToken("sys_id").ToString()

'Beispiel: snowResponse.SelectToken ("sys_id"). ToString ()


'Any internal fields or nested fields can be queried in the format:

'Alle internen Felder oder verschachtelten Felder können im folgenden Format
abgefragt werden:

'snowResponse.SelectToken("<parent1 variable name>").SelectToken("<Parent2 Or internal
field name>")....ToString()

'Example:" snowResponse.SelectToken("resolved_by").SelectToken("link").ToString()

'Beispiel: "snowResponse.SelectToken (" resolved_by "). SelectToken (" link ").
ToString ()


Dim sysID As String = snowResponse.SelectToken("sys_id").ToString()

Dim incnumber As String = snowResponse.SelectToken("number").ToString()

Dim status As String = snowResponse.SelectToken("state").ToString()

Dim ownerID As String = snowResponse.SelectToken("sys_updated_by").ToString()

Dim close_code As String = snowResponse.SelectToken("close_code").ToString()

Dim close_notes As String = snowResponse.SelectToken("close_notes").ToString()

Dim resolved_at As String = snowResponse.SelectToken("resolved_at").ToString()


'The value for the status variable should be modified to the value as configured at
the ServiceNow End

'Der Wert für die Statusvariable sollte auf den Wert geändert werden, der im
ServiceNow-Ende konfiguriert wurde


If status = "6" Then

'On the basis of the state, we can make the necessary decision on the
PersonWantsOrg entry.
```

'Auf der Grundlage des Staates können wir die notwendige Entscheidung über den Eintrag PersonWantsOrg treffen.

'Deutsche Übersetzung des folgenden Anrufs

'Beispiel: PWO.Custom.CallMethod("MakeDecision", "", True, "Ticket 123 wurde erfolgreich vom Benutzer admin geschlossen")


```
PWO.Custom.CallMethod("MakeDecision", "", True, "Ticket#" + incnumber + " was closed successfully by " + ownerID + " " + sysID +"")

PWO.PutValue("SNOWExt",True)

PWO.Save()

End If

Return True

Catch ex As Exception

Return False

End Try

End Function
```


Based on the configuration of values in the ServiceNow end, the value of the order state can be changed. For instance,

```
If status = "6" Then
```

'On the basis of the state, we can make the necessary decision on the PersonWantsOrg entry.

'Auf der Grundlage des Staates können wir die notwendige Entscheidung über den Eintrag PersonWantsOrg treffen.

'Deutsche Übersetzung des folgenden Anrufs

'Beispiel: PWO.Custom.CallMethod("MakeDecision", "", True, "Ticket 123 wurde erfolgreich vom Benutzer admin geschlossen")


```
PWO.Custom.CallMethod("MakeDecision", "", True, "Ticket#" + incnumber + " was closed successfully by " + ownerID + " " + sysID +"")

PWO.PutValue("SNOWExt",True)

PWO.Save()

End If
```

Here, a ServiceNow instance has the state "6" configured as close and assigned in ServiceNow. Hence, the custom call PWO.Custom.CallMethod("MakeDecision","", True,"Message") is set to true.

If a request is denied from the service now, then the custom call changes to PWO.Custom.CallMethod("MakeDecision","", True,"Message") is set to false.

Please ensure that the statement **PWO.PutValue("SNOWExt",True)** is present, after any decision making step as its value is used in other process chains.

The SnowResponse can be used to retrieve any field of the ServiceNow response. These retrieved values can be used to set the variables of a ticket. For example,

```
If status = "6" Then

'On the basis of the state, we can make the necessary decision on the
PersonWantsOrg entry.

'Auf der Grundlage des Staates können wir die notwendige Entscheidung über den Eintrag
PersonWantsOrg treffen.

'Deutsche Übersetzung des folgenden Anrufs

'Beispiel: PWO.Custom.CallMethod("MakeDecision", "", True, "Ticket 123 wurde
erfolgreich vom Benutzer admin geschlossen")


PWO.Custom.CallMethod("MakeDecision", "", True, "Ticket#" + incnumber + " was closed
successfully by " + ownerID + " " + sysID +"")

PWO.PutValue("SNOWExt",True)

PWO.Save()

End If
```

In this script, the message portion of the custom call is configurable. We have configured the message to display some text and values of the field "sys_udpated_by" and "sys_id". These values are retrieved before this If block as shown below:

Dim sysID As String = snowResponse.SelectToken("sys_id").ToString()

Dim ownerID As String = snowResponse.SelectToken("sys_updated_by").ToString()

Similarly more values can be retrieved with the syntax,

- *Dim testPropName As type = snowResponse.SelectToken("ServiceNow field name").ToString()*

For internal child fields,

- *Dim testPropName As type = snowResponse.SelectToken("ServiceNow parent name").SelectToken(child1)….ToString()*

These retrieved values can also be used to set some field of PersonWantsOrg entry like ReasonHead for instance.

General Syntax: *PWO.PutValue("<PersonWantsOrg field name>",testPropName)*

Example:

- *Dim closenotes As String = snowResponse.SelectToken("close_notes")*
- *PWO.PutValue("ReasonHead", closenotes)*

> ℹ️ **NOTE:**
>
> - **PWO.PutValue** is only applicable after the initialization in the script.
> - Script **SCN_UpdatingOneIMTicketStatus** cannot be modified directly. If further customizations are required on this script, new custom scripts must be created by copying this script's content and changes must be done on the new script. Change the function's name and use the same custom script name in the process step. The process chain referencing this script is **SCN_Check status of the ServiceNow ticket**. Modify the ScriptName in the internal process step "Updating the One Identity Manager with status of resolved tickets".

Constructing the Request for updating One Identity Manager status to ServiceNow ticket

The process chain **SCN_Update_1IM_ticket_status_to_ServiceNow** is responsible for constructing the response for ServiceNow update. This response is then sent to the ServiceNow end, through the process task **UpdateServiceNowStatus**. This task has a parameter **RequestBody** that takes the value from the preceding step, in the process chain, which calls the script **SCN_GetOrderValueStatus** that constructs the response. Below is the script with an example on how the script can be modified.

```
#If Not SCRIPTDEBUGGER

References Newtonsoft.Json.dll

#End If

'This script gets the OrderState value status from the PersonWantsOrg table with
reference of the UID_PWO sent from the calling step

'Dieses Skript ruft den OrderState-Wertstatus aus der Tabelle "PersonWantsOrg" mit der
Referenz der UID_PWO ab, die vom aufrufenden Schritt gesendet wurde

'On the basis of the OrderState value the necessary response is generated as explained
in this script further on

'Auf der Grundlage des OrderState-Werts wird die erforderliche Antwort generiert, wie
in diesem Skript weiter erläutert


Public Overridable Function SCN_GetOrderValueStatus(ByVal UID_PWO As String) As String

Dim PWO As ISingleDbObject = Connection.CreateSingle("PersonWantsOrg", UID_PWO)

Dim responseBodyStr As String = ""
```

```
Dim orderstate As String = PWO.GetValue("OrderState").String

Dim requestBody As Newtonsoft.Json.Linq.JObject = New Newtonsoft.Json.Linq.JObject()


'The Request JSON  for the response will have to constructed based on the value of the
OrderState in OneIM and

'the corresponding field values that are configured on the ServiceNow end.

'Der Anforderungs-JSON für die Antwort muss basierend auf dem Wert des OrderState
in OneIM und

'den entsprechenden Feldwerten erstellt werden, die auf dem ServiceNow-Ende
konfiguriert sind.

'To add any more fiels to the request JSON, add it in the format, requestBody.Add
("<fieldname>","<Value>")

'Um der Anfrage JSON weitere Felder hinzuzufügen, fügen Sie sie im Format
requestBody.Add ("<Feldname>", "<Wert>") hinzu.

'Example: requestBody.Add("close_code","Closed/Resolved by Caller")

'Beispiel: requestBody.Add ("close_code", "Closed / Resolved by Caller")


Select orderState
Case "Assigned"
requestBody.Add("close_code","Closed/Resolved by Caller")

requestBody.Add("state","7")

requestBody.Add("close_notes",PWO.GetValue("ReasonHead").String)

responseBodyStr = requestBody.ToString()

Case "Granted"

requestBody.Add("state","2")

responseBodyStr = requestBody.ToString()

Case "Dismissed"

requestBody.Add("close_code","Closed/Resolved by Caller")

requestBody.Add("state","7")

requestBody.Add("close_notes",PWO.GetValue("ReasonHead").String)

responseBodyStr = requestBody.ToString()

Case "OrderProduct"

requestBody.Add("state","2")

responseBodyStr = requestBody.ToString()
```

```
Case "Aborted"

requestBody.Add("close_code","Closed/Resolved by Caller")

requestBody.Add("state","7")

responseBodyStr = requestBody.ToString()

Case Else

responseBodyStr = ""

End Select

Return responseBodyStr

End Function
```

In the script, the **Cases** sections contain the responses for various **OrderState** values of the **PersonWantsOrg** entries.

For instance:

```
Case "Assigned"

requestBody.Add("close_code","Closed/Resolved by Caller")

requestBody.Add("state","7")

requestBody.Add("close_notes",PWO.GetValue("ReasonHead").String)

responseBodyStr = requestBody.ToString()
```

Here, the variable **requestBody** contains the request that is to be sent to ServiceNow. In this case, we are add the ServiceNow fields **close_code**, **state**, **close_notes**. These fields are mandatory for any ticket to be closed on ServiceNow for resolution. More fields can be added with the **requestBody** before the statement **responseBodyStr = requestBody.ToString()**.

- Syntax to add more fields: *requestBody.Add("<ServiceNow field name>","<Value>")*
- Example: *requestBody.Add("resoved_by",$DisplayPersonHead$)*

Here **$DisplayPersonHead$** is the value of the person taking a decision as configured in the approval work flow.

> ❶ NOTE: Script **SCN_GetOrderValueStatus** cannot be modified directly. If further customizations are required on this script, new custom scripts must be created by copying this script's content and then changes can be done on the new script. Change the function's name and use the same custom script name in the process step. The process chain referencing this script is **SCN_Update_1IM_ticket_status_to_ ServiceNow**. The internal process step where the ScriptName should be modified is **Script to process the OrderValue property of One Identity Manager**.

Troubleshooting

Issues related to the working of this module range from server related issues to job server latencies. Some of the issues that can affect the working of this module are mentioned below:

- Network issues connecting the ServiceNow instance with the Job Server handling ServiceNow tasks
- ServiceNow connectivity issues related to the instance's unavailability
- ServiceNow connectivity issues due to incorrect credentials
- ServiceNow connectivity issues due to controlled access over ServiceNow
- Job Server latency or downtime which would affect the ticket creation for the PersonWantsOrg requests

The triggering of ServiceNow process chains depend on the One Identity Manager database changes.

🛈 NOTE: If ServiceNow activities do not work as expected, job server logs must be observed and validated for errors.

Appendix

Sample ServiceNow Response

The ServiceNow response for the tickets has many attributes which can be retrieved in GET scenarios and can be updated for PUT/POST scenarios. Navigate to https://docs.servicenow.com/bundle/geneva-servicenow-platform/page/integrate/inbound_rest/reference/r_TableAPI-GET.html response and view a sample response from ServiceNow.

One Identity Manager uses the following sets of attributes for Ticket Management / Incident Management:

- sys_id - This attribute contains the ServiceNow system id that is used in processing all the process chains configured for ServiceNow

- state - This attribute is responsible for tracking the ticket status and updating One Identity Manager accordingly

- incnumber - This attribute holds the incident number of the ticket

Contacting us

For sales or other inquiries, visit https://www.oneidentity.com/company/contact-us.aspx or call +1-800-306-9329.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product