One Identity by Quest

One Identity Manager and ServiceNow Integration 8.1.5

Administration Guide for Connecting to ServiceNow

Contents

ServiceNow Module Overview

The One Identity Manager IT Shop allows users to request company resources such as applications, system roles, or group membership as well as non-IT resources such as mobile telephones or keys. The integration of the IT Shop in ServiceNow enables employees to request products directly from ServiceNow. This Integration also provides governance over IT request, by ensuring that clear audit trails and controls are in place to meet security and compliance requirements.

Using the One Identity Manager for Service Catalog App users can raise IT Shop requests with the fulfillment being handled by One Identity Manager. Having passed through an approval workflow in ServiceNow, requests then proceed to One Identity Manager. Requests are automatically created in One Identity Manager and the defined workflow starts. This provides the ability to leverage both workflow engines but also allows for self-service. ServiceNow users need an employee record in One Identity Manager to submit requests. Employees are synchronized into a ServiceNow application table using the One Identity Manager Application Server and are matched to ServiceNow Users (sysuser) using the central account (optionally a configurable property) and the UserID of the ServiceNow User.

An application-specific approval workflow, containing application-specific steps, is provided for ServiceNow and can be modified to meet your requirements. In One Identity Manager, the requests are processed by a flexible policy-based approval process. The request history makes it possible to follow who requested which company resource or hierarchical role and when it was requested, renewed, or canceled. By default, a product request by a user will require approval by the user's manager. If the user has a manager, the approval goes to the manager and would be directed to a configurable group of fallback approvers if the user does not have a manager configured.

ServiceNow One Identity Manager App user can raise an IT request and the fulfillment of the request is handled by the approval workflow in ServiceNow. Based on the request approval outcome in ServiceNow, the request then proceeds to One Identity Manager by automation of request creation and proceeded by the kind of workflow attached to the service item. One Identity Manager data is read and updated by the integration using One Identity Manager Application Server. One Identity Manager Employees are synced into the ServiceNow custom users table based on the match between the central account (and optionally configured custom property) and the UserID of the ServiceNow User (sysusers), post which the ITShop request can be raised in ServiceNow.

The requests are processed by a flexible policy-based approval process. Introducing IT Shop avoids time-consuming demands within the company and reduces the administration

effort. Requests follow a defined approval process which decides whether a product may be assigned or not. For Example, a product requested by the user goes for an approval workflow, if the user has a manager the approval goes to the manager and if the user doesn't have any manager then it goes to fallback approver.

Installation

The following sections described the installation of ServiceNow module and One Identity Manager for service catalog as required.

Installation of ServiceNow Module

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

Installation of One Identity Manager for Service Catalog

Steps to install One Identity Manager for Service Catalog:

1. Install the One Identity Manager Service Catalog App and make it available on your instance

2. Navigate to **System Applications** | **All Available Applications** | **All**.

3. Find the application using the search bar

4. Click Install.

5. In the Application installation dialog box, review the application dependencies.

   Dependent plugins and applications are listed if they will be installed, are currently installed, or need to be installed. If there are any plugins or applications that need to be installed, you must install them before you can install the ServiceNow Store application.

6. Click Install.

NOTE: This is an optional step and is only required if One Identity Manager for Service Catalog is required

Managing ServiceNow Incidents from One Identity Manager

This section provides details of integration of ServiceNow with One Identity Manager for managing ServiceNow incidents.

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
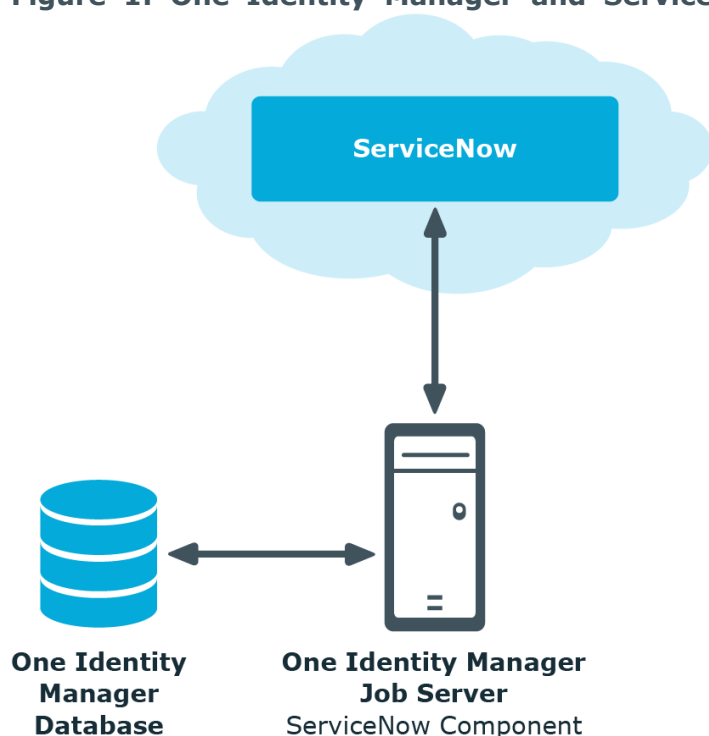
The integration solution uses the following process chains to create and track service tickets:

- 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.

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 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 ServiceNow 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 Designer.

Enabling service categories for ServiceNow

To enable service categories for ServiceNow

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

2. On the Workflow editor, go to **Toolbox**, under **Workflow** and click **Edit**.

3. 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 editor.

4. 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.

5. 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.

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

7. On the toolbar, click **Save** to save the approval workflow.

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

9. 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.

10. 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 web 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 | QBM| Install | bin** and run the **WebDesigner.ConfigFileEditor.exe** 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 **SCN_Check_status_of_the_ServiceNow_ticket** process 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 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 **ServiceNow 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 **SCN_Create ServiceNow ticket** process 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 **SCN_Update 1IM status to ServiceNow** process is triggered if any change is made on the status of the

request. This process ensures that the JSON response is sent to the ServiceNow end, and the associated ticket is updated in ServiceNow.

> NOTE: If the **SCN_Update 1IM status to ServiceNow** process fails, it is recommended to create a new MailComponent process 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 request , 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 **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 **Create ServiceNow 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 **SCN_Create ServiceNow Ticket** process 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, 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 SNOW Response.

SNOW Response 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 SNOW Response value is used in the script **SCN_UpdatingOneIMTicketStatus**. This is first parsed to a NewtonSoft Object 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 One Identity Manager.
```

```
'Dieses Skript wird zum Aktualisieren des Status vom ServiceNow-Ticket zum
entsprechenden PersonWantsOrg-Eintrag in One Identity Manager 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 One Identity Manager
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.
```

```
'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 ServiceNow, 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 processes.

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)*

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 script name in the following 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 **SCN_Update_1IM_ticket_status_to_ServiceNow** process 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, 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 One Identity Manager 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.

ⓘ 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. process that references this script is **SCN_Update_1IM_ticket_status_to_ServiceNow**. The internal process step where the script name should be modified is **Script to process the OrderValue property of One Identity Manager**.

One Identity Manager for Service Catalog

This section provides details of integration of One Identity Manager with Service catalog in ServiceNow.

Configuring ServiceNow's One Identity Manager Catalog Application

This section elaborates configuration of One Identity Manager service catalog app for integration with One Identity Manager.

Adding Catalog Item to the Catalog Page

After the installation of One Identity Manager ServiceNow App, it needs to be added to the service catalog page.

Steps to add One Identity Manager ServiceNow App to catalog page:

1. Navigate to the **Self-Service** | **Service Catalog** in your instance.
2. **Add content** | Search for **One Identity Manager for Service Catalog** from the categories
3. Click **Add** here.
4. The Catalog Item will then be available in the Catalog Page to place IT Shop requests from ServiceNow.

Configuration Parameters

The details related to configuration parameters in the One Identity Manager ServiceNow App are listed below Configuration parameters can be found under One Identity Manager ServiceNow App in the Application navigator. Sysadmin/Appadmin would be able to configure these parameters. Initially when the One Identity Manager Service App is configured for the first time the scheduled job must be executed manually once for the configuration parameters to be available for the user to edit.

To execute the Scheduled job follow the below steps:

1. Open the ServiceNow instance.
2. Navigate to **System Definition** | **Scheduled jobs**.
3. Search for **InitializeConfigurationParametersAndLoadData** and select it.
4. Click on **Execute** button to execute the background job and initialize the configuration parameters

Table 1: Configuration Parameters

Config Parameter Name	Config Parameter Description
add_OneIM_ managers_to_ approver_role	If set to true, Managers configured in One Identity Manager are added to the ServiceNow **approver_user role**.
compliance_officer	The group of the compliance officers which will approve, if a request would lead to a SoD conflict.
delta_load_data_ from_oneim_ server_persons	If true, loads employee data from One Identity Manager to ServiceNow during a delta sync .
delta_xdateup- dated_accproduct	Highest XDateUpdated for Service Item entities, to be used for the next delta run. This value is automatically calculated and set after Full / Delta sync.
delta_xdateup- dated_accpro- ductgroup	Highest XDateUpdated for Service Categories entities, to be used for the next delta run. This value is automatically calculated and set after each Full / Delta synchronization.
delta_xdateup- dated_person	Highest XDateUpdated date for Person entities from the last synchronization. To be used for the next delta run. This value is automatically calculated and set after each Full / Delta synchronization.
fallback_approver	The approval is sent to the fallback approver group if no manager is available .
job_execution_	Background Job execution status (Ready/Running). **Ready**

ONE IDENTITY
by Quest

Config Parameter Name	Config Parameter Description
status	implies the job is not executing and **Running** implies job is executing.
job_load_data_ from_oneim_ server_delta_load	Scheduled Job: Perform delta load from One Identity Manager during scheduled run? (true | false)
job_load_data_ from_oneim_ server_full_load	Scheduled Job: Perform full load from One Identity Manager? (true | false)
log_level	Info = show info, warning, and error log messages debug = show info, warning, error, and debug log messages warn = show error and warning log messages error = show error log messages only
oneim_api_ retrythreshold	The number of times the retry mechanism should be executed in case of failures.
oneim_request_ validity_default	IT Shop request validity. If the validity fields (valid from and valid until) are empty, the validity will be set to default validity (in days).
oneim_rest_ endpoint_url	REST API Endpoint to the One Identity Manager Application Server
oneim_rest_ pagelimit	The number of items that can be fetched per page during the API call to One Identity Manager App Server.
oneim_rest_ password	Password of the service user for the REST API's Endpoint of the One Identity Manager Application Server.
oneim_rest_ username	Username of the service user of REST API's Endpoint to the One Identity Manager Application Server
oneim_to_servi- cenow_user_ matching_attribute	This attribute is used to configure an alternate property other than central account that can be used to match the Person to sysusers in ServiceNow. This is an optional attribute.
page_load_data_ from_oneim_ server_delta_load	If true, perform a delta load from One Identity Manager during page load .
page_load_data_ from_oneim_ server_full_load	If true, perform a full load from One Identity Manager to during page load .
delta_load_data_ from_oneim_	Get new Service Items from One Identity Manager on page load.

Config Parameter Name	Config Parameter Description
server_service_ items	
perform_manager_ approval	This parameter is used to determine whether to skip ServiceNow Manager approval completely .
perform_sod_check	Should a SoD check be performed during request submission? (true | false)
manager_ approval_author- itative_source	Determines whether ServiceNow or One Identity Manager is the authoritative source for managers.
truncate_user_ accproductgroup_ accproduct	When the scheduled job executes and this parameter is set to true, the tables x_oni_oneim_addon_accproduct_user and x_ oni_oneim_addon_accproductgroup_user are truncated in order to reduce the overhead of maintaining large data in custom tables.

These configuration parameters are required for importing data into ServiceNow:

- oneim_rest_password
- oneim_rest_username
- oneim_rest_endpoint_url

NOTE: The password must be entered in the config value encrypted field, other parameters can be entered into config value.

Initial load from One Identity Manager to ServiceNow

Once the ServiceNow App is installed and configured, the One Identity Manager entities including **Person**, **Service Category** and **Service Items** must be synchronized to the **ServiceNow** instance. The import of data can be done using a scheduled job.

Scheduled execution

The One Identity Manager for Service Catalog App includes a scheduled server script that imports the One Identity Manager Employees and IT Shop Categories and Items into ServiceNow. You can find this by navigating to **System Definition** | **Scheduled jobs** in a ServiceNow instance and searching for **InitializeConfigurationParametersAndLoadData**.

- This server scheduled script executes in the background at the specified time interval.
- Run and Time fields can be customized to schedule the job.
- By default the scheduled job runs at 12:00:00 GMT daily.

NOTE: One Identity recommends to have the schedule script running during non peak hours.

Data Synchronization from One Identity Manager to ServiceNow

This section explains about the synchronization of data from One Identity Manager to ServiceNow catalog integration.

Matching One Identity Manager Employees to ServiceNow users

One Identity Manager employees are matched to ServiceNow users by comparing Identity Manager Employee's Central Account to ServiceNow user's UserID. If a match cannot be found and configuration parameter **oneim_to_servicenow_user_matching_attribute** has been configured, matching is performed by comparing the One Identity Manager custom property to the ServiceNow UserID.

NOTE: The source field of the ServiceNow user is automatically set to OneIdentityManager during Synchronization. This should not be changed.

Data Synchronization use cases

There are two ways data can be synchronized from One Identity Manager to ServiceNow.

- **Full sync**: This means that all data will be loaded from One Identity Manager.
- **Delta sync**: This means that all data will be loaded from One Identity Manager, which was added or updated after the last synchronization date. These configuration parameters are updated after every synchronization.

  For example, only those employees that are created/updated after the date defined in the configuration parameter "delta_xdateupdated_person" will be imported. This reduces the import duration.

Performing a full synchronization through scheduled job

Scheduled Script Executions: **InitializeConfigurationParametersAndLoadData**.

The configuration parameter **job_load_data_from_oneim_server_full_load** determines if the scheduled job should perform a full synchronization. This parameter

takes a boolean value (default value is **true**) and setting the value to **true** would enable a full synchronization.

> NOTE: One Identity recommends performing Delta load of users and service items through a scheduled background job on a daily basis during non peak hours. Full load of users and service items could be performed once a month or according to customer requirements during non peak hours.

Performing delta synchronization through scheduled job

Scheduled Script Executions: **InitializeConfigurationParametersAndLoadData**.

The configuration parameter **job_load_data_from_oneim_server_delta_load** is used to configure delta synchronization by the scheduled job service. This parameter takes a Boolean value (default value is **false**) and setting the value to **true** would cause a delta synchronization to be performed if full synchronization is not enabled.

Once the delta synchronization has been enabled, configure the following additional configuration parameters that specify what entities will be delta synchronized

Additional delta synchronization configuration parameters

- delta_load_data_from_oneim_server_persons: If set to true, One Identity Manager persons would be delta synchronized in to ServiceNow depending on the value of the configuration parameter **delta_xdateupdated_person**.
- delta_load_data_from_oneim_server_service_items: If set to true, One Identity Manager Service Items would be delta synchronized in to ServiceNow depending on the value of the configuration parameter **delta_xdateupdated_accproductgroup**.

Performing full synchronization through Catalog Page

The configuration parameter **page_load_data_from_oneim_server_full_load** is used for performing full synchronization on the page load. This parameter takes Boolean value (default value is **false**) and setting the value to true would perform a full sync.

> NOTE: Full synchronization should not be performed on the catalog page since it will lead to significant performance degradation. Full synchronization should be done only through the job service described earlier. Full synchronization through catalog page should be used only for testing purposes.

Performing delta synchronization through Catalog Page

The configuration parameter **page_load_data_from_oneim_server_delta_load** is to configure delta synchronization on the catalog page. This parameter takes a Boolean value and setting the value to **true** would perform a delta synchronization if the catalog page full synchronization is not enabled.

Once the catalog page delta synchronization configuration parameter is configured, the additional delta synchronization configuration parameters **delta_load_data_from_ oneim_server_persons**, **delta_load_data_from_oneim_server_service_items** also need to be configured to define which objects should be delta synchronized. These parameters have already been explained earlier.

Roles and Permissions

Details of the roles that are currently supported by the **One Identity Manager for Service Catalog** App are explained below.

- **x_oni_oneim_addon.admin** – This is the **One Identity Manager for Service Catalog** App Administrator role. Just like the SysAdmin, these users can request service items for any user that has a matching employee record in One Identity Manager. It is the responsibility of the SysAdmin to assign this role to appropriate users. Users with this Role would be able to view the application in the application navigator and will have Read/Write access to all the application tables.

- **x_oni_oneim_addon.businessuser** – This is the **One Identity Manager for Service Catalog** application business user role. These users can request service items only for themselves. All users synchronized into ServiceNow from One Identity Manager will be assigned to this role.

Approver roles

Once an IT shop request is created, it follows the defined approval process. If manager approval is enabled in configuration parameters, the request is routed to the manager for approval. The manager needs an appropriate role such as the approver_user role, to be able to approve or reject the IT Shop request.

Adding approval role for One Identity Manager managers

Once a synchronization operation completes One Identity Manager managers optionally could be added to a configured approval role. The configuration parameters for automatically adding One Identity Manager managers approver role are:

- **add_OneIM_managers_to_approver_role**: Boolean value (true/false) that determines whether One Identity Manager Managers will be added to the ServiceNow approver role approver_user.

NOTE: The role could be chargeable. Consult a ServiceNow representative regarding cost involved before enabling this configuration parameter.

One Identity Manager ServiceNow App Tables

One Identity Manager ServiceNow application uses custom tables to store the application related configurations and data that are synchronized from the One Identity Manager.

Details of the tables are summarized below.

Configuration Parameters

This table is used to store the One Identity Manager ServiceNow application configuration parameters that can be edited according to the business requirement. This table is only visible to the users with the admin role.

The following table contains a summary of all the custom columns added.

Column Name	Descriptions
Config param	This column defines the name of the configuration parameter
Config desc	This column provides the configuration parameter description
Config value	This column is used to enter the value for the config parameter
Config value encrypted	This column is used to enter sensitive data for security reasons. For example password

IT Shop Service Category

Service Categories in One Identity Manager ITShop are synchronized from One Identity Manager to ServiceNow into this custom table. This table is only accessible to the users with the admin role.

The following table contains a summary of all the custom columns added.

Column Name	Descriptions
Service Category	This column provides the name of the service category in the IT shop service catalog
Description	This column provides the description relating to the respective service catalog given in the One Identity Manager
Unique ID	This column stores the GUID of the service catalog present in One Identity Manager
XobjectKey	This column stores the XObjectKey for the respective Service catalog in One Identity Manager

IT Shop Service Category – User

Mapping between the One Identity Manager ITShop ServiceCategory and Users are synchronized into this table. This table is only accessible to the users with the admin role.

The following table contains a summary of all the custom columns added.

Column Name	Descriptions
Service Category	This column provides the name of the service category in the IT shop service catalog
User	Name of the user that has the resources

IT Shop Service Items

IT Shop services created under the IT Shop Service Category are synchronized into this table in ServiceNow and are used for assigning the service items in the ServiceNow catalog page. This table is only visible to the users with the admin role.

The following table contains a summary of all the custom columns added.

Column Name	Descriptions
Service Item	Name of the Service Item created in the One Identity Manager IT Shop
Unique ID	GUID of the Service Item created in One Identity Manager
Service Category	Name of the service category under which the Service Item is created in One Identity Manager
UID_ ITShopOrg	GUID of IT Shop Org present in One Identity Manager
XObjectKey	Unique XObject Key Present in One Identity Manager

IT Shop Service Items - User

Mapping between the One Identity Manager ITShop ServiceItems and Users are synchronized into this table. This table is only visible to the users with the admin role.

The following table contains a summary of all the custom columns added.

Column Name	Descriptions
Service Item	Name of the Service Item created in the One Identity Manager IT Shop
Service Category	Name of the service category under which the Service Item is created in One Identity Manager
User	Name of the user that has the resources

Shopping Cart Order

All the request orders that are created for a user on the ServiceNow catalog page are stored here. This table is only visible to the users with the admin role.

The following table contains a summary of all the custom columns added.

Column Name	Descriptions
UID_ShoppingCartOrder	UID_ShoppingCartOrder present in the ShopCartOrder table in One Identity Manager is synchronized in this column
Request	This column provides the Request Number for the request raised through the One Identity Manager for ServiceNow catalog page

Users

All the Employees from One Identity Manager are synchronized into ServiceNow to this table, if the useraccount exists for the Employee in the sysusers table. Also, the userid for ServiceNow account should match the Central Account / the CustomProperty value. This table is only visible to the users with the admin role.

The following table contains a summary of all the custom columns added.

Column Name	Descriptions
Firstname	First name of the employee in One Identity Manager
Lastname	Last name of the employee in One Identity Manager
Displayname	Display name of the employee in One Identity Manager
ServiceNow User ID	ServiceNow user ID
ServiceNow User Name	ServiceNow UserName
CustomProperty	The custom property in the One Identity Person table. Optionally this ID can also be used to match One Identity Manager employees to ServiceNow users
UID Person	GUID of the employee in One Identity Manager person table
UID_PersonHead	GUID of the manager present in the One Identity Manager person table
XObjectKey	XObject key present for all the employees in the One Identity Manager person table

NOTE: As these columns are used in various scripts, the column/table names should not be modified as they will lead to exceptions.

Raising a request and approval workflow

One Identity Manager ServiceNow Application allows users that are assigned admin role/sysadmin/businessuser to request company resources such as applications, system roles, or group membership as well as non-IT resources such as mobile telephones or keys.

The resources are requested using the IT Shop from the ServiceNow catalog page. The detailed procedure to request an IT Shop items is explained below.

To request an IT Shop item from ServiceNow Catalog page:

1. From the ServiceNow instance portal navigate to the Catalog page.
2. Search for One Identity Manager for Service Catalog.
3. Enter the Required details, and click on the submit button

NOTE:

- Fetch specific service category for a user using key search: If a particular service category is not available in the picker on a search, click on the refresh button below in order to sync the categories from One Identity Manager Application Server. Once the categories are refreshed the user can select the specific category using the picker.

- Wildcard search of service categories for a selected user: In order to fetch all the service categories for the selected user from the One Identity Manager Application server type '**' in the service category picker and then click on the refresh button below. Once the categories are refreshed the user can view all the categories fetched by entering '*' in the picker.

- Fetch specific service item for a selected service category and user using key search: If a particular service item for a selected service category is not available in the picker on a search, click on the refresh button below in order to sync the service item from One Identity Manager Application Server. Once the service items are refreshed the user can select the specific item using the picker.

Request is submitted and processed based on the configuration combinations and approval workflow.

Process overview

Once an IT Shop request is raised, it follows a defined approval process which decides whether the request is be approved or rejected.

Figure 2: Approval workflow process
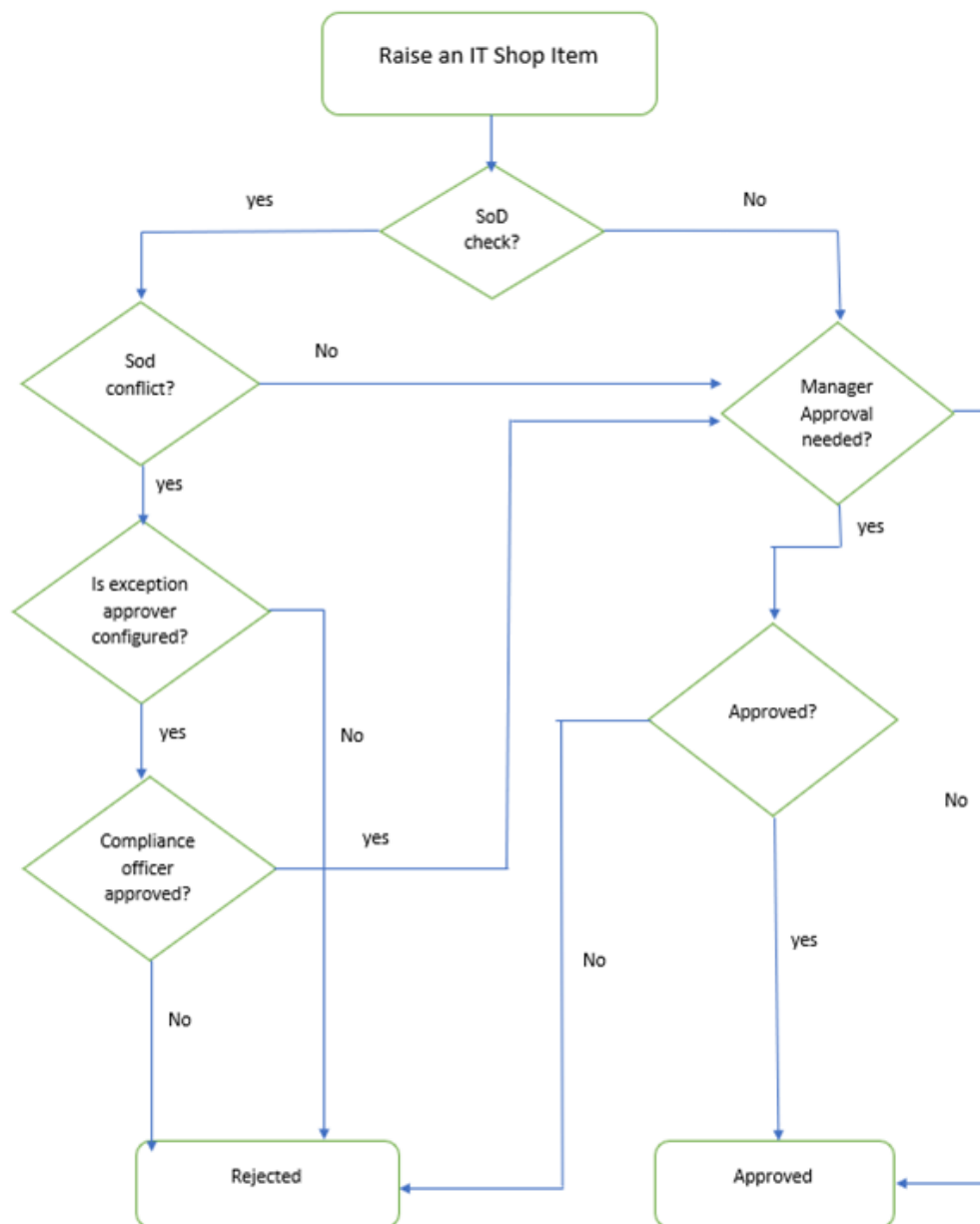
ServiceNow Approval

A request raised on ServiceNow is routed to the manager for approval or follows self-service approval depending on how the configuration parameters are configured. If manager approval is configured, each requested item will be available for separate approval/rejection, provided that SOD check is not enabled. If SOD check is enabled, all requested items need to be approved/rejected in a single operation.

Manager Approval

If manager approval is enabled, the request is routed to user's ServiceNow/One Identity Manager's manager for approval depending on the configuration parameter. Configure the following configuration parameters described below

Config name	Value
perform_manager_approval	true
manager_approval_authoritative_ source	SNOW / ONEIM
fallback_approver	"Fallback approver name"

If **manager_approval_authoritative_source** has been configured to SNOW, the request will be routed to user's ServiceNow manager and if one does not exist, it is routed to the configured fallback approver.

If **manager_approval_authoritative_source** has been configured to ONE IDENTITY MANAGER, the request will be routed to user's One Identity Manager's manager and if one does not exist, it is routed to the configured fallback approver.

NOTE: If the authoritative source is ServiceNow then system admin should make sure that the appropriate manager has approver role.

Self-Service approval in ServiceNow

To enable self-service approval in ServiceNow, configure the following configuration parameters with the value specified

Config name	Value
perform_manager_approval	false

Now the user requests will be automatically approved.

SOD cases

SOD rules configured in One Identity Manager can be checked and validated against at ServiceNow end by enabling the configuration parameter perform_sod_check (set the configuration parameter to true). SOD use cases are summarized below:

- **No SOD conflict for any of the requested item**: The request is routed to the configured manager/fallback approver/self-service approval is performed.

- **SOD Conflict for some of the requested items and exception approver has been configured in the One Identity Manager SOD Rule**: The request is routed to the compliance officer configured in ServiceNow (Configuration parameter: compliance_officer). If the compliance officer approves the request, the request is then routed to the configured manager/fallback approver/self-service approval is performed. If compliance officer rejects, the request is rejected

- **SOD Conflict for some of the requested items and exception approver has not been configured in the One Identity Manager SOD Rule**: The request is automatically canceled.

One Identity Manager Approval

Once the IT Shop Item is approved in the One Identity ServiceNow application, the request is then processed by the defined approval process in One Identity manager. Optionally ITShop approval policy could be configured in such a way that self-service approval takes place when the request has been raised and approved in ServiceNow while request raised from One Identity Manager goes over the regular approval process. This way approvals do not need to take place multiple times for request raised from ServiceNow.

For more information on IT Shop Request approval process please refer to the *Identity Manager 8.1 - IT Shop Administration Guide*.

Logging

Log level can be configured using the configuration parameter log_level. The different log levels and their description are explained in below table. The default log_level is info.

Log Level Description	Log Level Description
error	Logs events that might still allow the application to continue running. Setting the log level for an application to error generates error messages only, but does not generate warn, info, or debug messages.
warn	Logs potentially harmful events. Setting the log level for an application to warn generates error and warn messages but does not generate error or debug messages.
info	Logs informational messages that describe the progress of the application. Setting the log level for an application to info generates info, warn, and error messages, but does not generate debug messages.
debug	Logs informational events that are useful for debugging an application. Setting the log level for an application to debug generates info, warn, error, and debug messages.

Troubleshooting

This section covers the troubleshooting guidelines for the One Identity Manager and ServiceNow integration.

One Identity Manager for Service Catalog

Unable to load data in to ServiceNow from One Identity Manager

Test to make sure the One Identity Manager App server is accessible from ServiceNow. After executing the Initial Synchronization in section (see Initial load from One Identity Manager to ServiceNow), in case the data is not loaded into the application tables, verify the below steps:

1. Navigate to the application logs to check for any error related to the connectivity between ServiceNow and One Identity Manager application server

2. Verify that One Identity Manager App server is accessible through the browser using the same credentials and Application server URL provided in the configuration

Catalog Page load takes long time to load

Make sure the catalog page full synchronization configuration parameter (page_load_data_ from_oneim_server_full_load) has been turned off.

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. Please refer Table API documentation at https://docs.servicenow.com to see 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 processes 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

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit https://www.oneidentity.com/company/contact-us.aspx.

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 at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product