



One Identity Manager and ServiceNow
Integration 9.3.1

Administration Guide for Connecting
to ServiceNow

Copyright 2025 One Identity LLC.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of One Identity LLC .

The information in this document is provided in connection with One Identity products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of One Identity LLC products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, ONE IDENTITY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ONE IDENTITY BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ONE IDENTITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. One Identity makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. One Identity does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

One Identity LLC.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (<http://www.OneIdentity.com>) for regional and international office information.



Patents

One Identity is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at <http://www.OneIdentity.com/legal/patents.aspx>.

Trademarks

One Identity and the One Identity logo are trademarks and registered trademarks of One Identity LLC. in the U.S.A. and other countries. For a complete list of One Identity trademarks, please visit our website at www.OneIdentity.com/legal. All other trademarks are the property of their respective owners.

Legend

-  **WARNING:** A WARNING icon highlights a potential risk of bodily injury or property damage, for which industry-standard safety precautions are advised. This icon is often associated with electrical hazards related to hardware.
-  **CAUTION:** A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

Contents

| | |
|--|-----------|
| ServiceNow Module Overview | 6 |
| Installation | 8 |
| Installation of ServiceNow Module | 8 |
| Installation of One Identity Manager for Service Catalog | 8 |
| Managing ServiceNow Incidents from One Identity Manager | 10 |
| Managing ServiceNow | 10 |
| Technical Overview | 11 |
| Configuring ServiceNow module | 12 |
| Configuring ServiceNow Connection Parameters | 12 |
| Enabling service categories for ServiceNow | 13 |
| Configuring One Identity Manager Custom IT Shop for ServiceNow module | 13 |
| Creating approval policies for external approval | 13 |
| Assigning external approval policy to IT Shop | 14 |
| Using ServiceNow as a One-Stop Shop | 14 |
| Configuring ServiceNow web portal | 15 |
| Configuring One Identity Manager Web Portal Authentication Modules | 16 |
| Logging out from ServiceNow | 17 |
| Use Case Scenarios | 17 |
| Use Case Scenario with ServiceNow as Master | 18 |
| Use Case Scenario with One Identity Manager as Master | 18 |
| Customizing ServiceNow module | 19 |
| Creating ServiceNow Ticket – Adding more attributes to the ticket creation | 19 |
| Manipulating the response from ServiceNow | 19 |
| Constructing the Request for updating One Identity Manager status to ServiceNow ticket | 24 |
| One Identity Manager for Service Catalog | 28 |
| Configuring ServiceNow's One Identity Manager Catalog Application | 28 |
| Adding Catalog Item to the Catalog Page | 28 |
| Configuration Parameters | 29 |
| Initial load from One Identity Manager to ServiceNow | 33 |

| | |
|---|-----------|
| Scheduled execution | 34 |
| Data Synchronization from One Identity Manager to ServiceNow | 34 |
| Roles and Permissions | 35 |
| Schedule job OneIdentity Manager user permissions required | 36 |
| Approver roles | 36 |
| Adding approval role for One Identity Manager managers | 36 |
| One Identity Manager ServiceNow App Tables | 37 |
| Login with SSO | 41 |
| Setting Up Cryptographic Keys for the One Identity Manager Cryptographic Module | 41 |
| Raising a request and approval workflow | 42 |
| Adding more information to Approvers | 44 |
| Steps to Configure the OneIM Approval Record Widget | 44 |
| Enabling Indirect Reportees for Request Creation | 45 |
| Process overview | 45 |
| ServiceNow Approval | 46 |
| Manager Approval | 46 |
| Self-Service approval in ServiceNow | 47 |
| SOD cases | 47 |
| One Identity Manager Approval | 48 |
| Customizing the Application | 48 |
| One Identity Manager Login Logging Customization Example | 48 |
| Create One Identity Manager employee from ServiceNow | 49 |
| Detailed explanation of the fields | 50 |
| Enabling One Identity Manager's Shopping Cart in ServiceNow | 52 |
| One Identity Manager, Microsoft Entra ID, and ServiceNow MultiSSO Integration (OIDC) | 55 |
| Prerequisites | 55 |
| Register Application in Azure AD | 55 |
| Microsoft Graph API Permissions | 56 |
| Create a Client Secret in Azure | 56 |
| Assign Users to the Azure AD Enterprise Application | 57 |
| Gather Azure App Info | 57 |
| Configure OAuth in One Identity Manager | 57 |
| Enable MultiSSO Plugin in ServiceNow | 58 |
| Configure the OIDC Identity Provider in ServiceNow | 59 |

| | |
|--|-----------|
| Testing the Integration | 59 |
| Troubleshooting Tips | 60 |
| ServiceNow Mid Server Support | 62 |
| Creating a MID Server Administrator User | 62 |
| Creating a Service Account User | 62 |
| Installing Mid Server in ServiceNow | 63 |
| Validating MID Server in ServiceNow | 63 |
| Testing the MID Server | 64 |
| How to use MID Server | 64 |
| Logging | 65 |
| Troubleshooting | 66 |
| One Identity Manager for Service Catalog | 66 |
| About us | 68 |
| Contacting us | 68 |
| Technical support resources | 68 |

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 integration utilizes the One Identity Manager Application Server and API Server to read and update data within the One Identity Manager database. The Application Server is

employed to retrieve information pertaining to users, service categories, and service items. Concurrently, the API Server is tasked with accessing data related to request properties and facilitating the creation of actual requests within the One Identity Manager database.

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 (Store version - 2.0.0) 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.

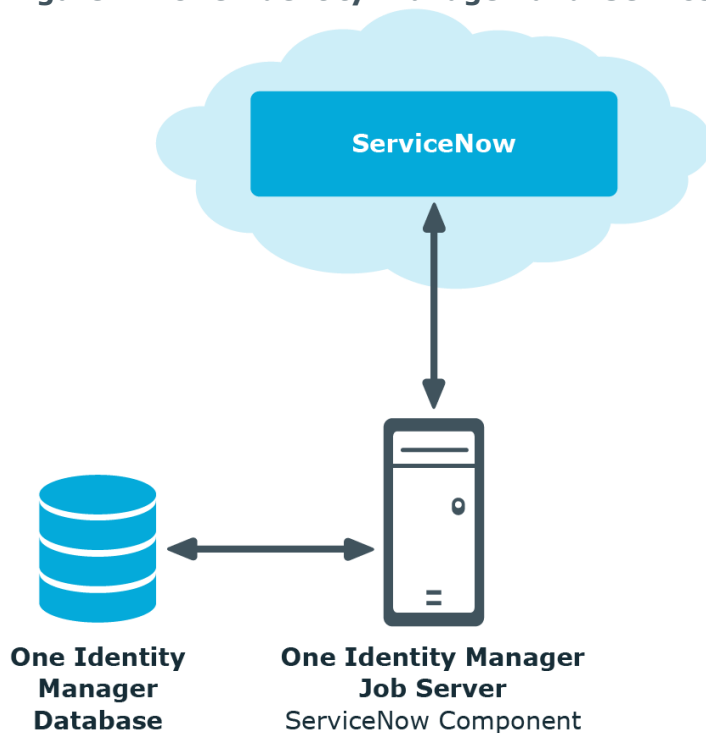
Another form of integration between One Identity Manager and ServiceNow provides the control to One Identity Manager to act as the master. Tickets are created for service items

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 policies

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 Manager 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.

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 Admin-
istration 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-Admin-
istrationshandbuch.

'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: PW0.Custom.CallMethod("MakeDecision", "", True, "Ticket 123 wurde erfolgreich vom Benutzer admin geschlossen")
```

```
PW0.Custom.CallMethod("MakeDecision", "", True, "Ticket#" + incnumber + " was closed successfully by " + ownerID + " " + sysID + "")
```

```
PW0.PutValue("SNOWExt", True)
```

```
PW0.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: PW0.Custom.CallMethod("MakeDecision", "", True, "Ticket 123 wurde erfolgreich vom Benutzer admin geschlossen")
```

```
PW0.Custom.CallMethod("MakeDecision", "", True, "Ticket#" + incnumber + " was closed successfully by " + ownerID + " " + sysID + "")
```

```
PW0.PutValue("SNOWExt", True)
```

```
PW0.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# + 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_updated_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 be 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 fields 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",PW0.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",PW0.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", PW0.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 | Add OneIM Managers to ServiceNow's approver_user role so that they would be able to approve the ServiceNow request assigned to them? (true false) |
| allow_indirect_reportees | This parameter is used to determine whether a user can raise a request for indirect reportees or not. (true false) |
| Auth_type | Specifies the type of authentication. Expected values: <ul style="list-style-type: none">• Password – Only the standard login page appears.• Both – Users can log in using either IdP SSO or password credentials.• OAuth – Application is authenticated with the user's IdP credentials and login is automatic if all IdP parameters are set. |
| compliance_officer | The group of the compliance officers which will approve, if a request would lead to a SoD conflict. |
| default_employee_type | Set the default Employee type for the Person OnBoarding form. List of possible employee type: <ul style="list-style-type: none">• Employee• Other |

| Config Parameter Name | Config Parameter Description |
|--|---|
| | <ul style="list-style-type: none"> • Apprentice • Consultant • Contractor • Customer • Partner |
| delta_load_data_from_oneim_server_organizations | If true, loads organisations data from One Identity Manager to ServiceNow during a delta sync |
| delta_load_data_from_oneim_server_persons | If true, loads employee data from One Identity Manager to ServiceNow during a delta sync . |
| delta_load_data_from_oneim_server_service_categories | If true, loads service categories data from One Identity Manager to ServiceNow during a delta sync. |
| delta_xdateupdated_accproductgroup | 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_xdateupdated_organizations | Highest XDateUpdated date for Organization entities from the last synchronization. To be used for next delta run. This value is automatically calculated and set after each Full/Delta synchronization. |
| delta_xdateupdated_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. |
| employee_type | This parameter is used to determine what kind of employees are fetched from OneIM. If you want more than one of a kind, then provide comma separated values. Example- contractor, employee |
| fallback_approver | The approval is sent to the fallback approver group if no manager is available . |
| idp_auth_url | Specifies the authorization URL of the Identity Provider (IdP). Used to redirect users for login and consent. |

| Config Parameter Name | Config Parameter Description |
|--|--|
| idp_client_id | The client (application) ID assigned by the IdP when the application is registered. Identifies the app during authentication. |
| idp_client_secret | The client secret issued by the IdP. Used like a password by the application to authenticate with the IdP. |
| idp_label | A readable label for the IdP. It can have a customized name (e.g., "SSO") displayed in the UI. (Default value is - IDP) |
| idp_redirect_uri | This must match the redirect URI registered in the IdP. Example: https://<instance>.service-now.com/sp?id=OneIMLoginPage |
| idp_tenant_id | The unique tenant (directory) ID in the IdP (For e.g., Azure AD: Directory ID). Used to scope authentication requests. |
| idp_token_endpoint | The endpoint used to exchange authorization codes or refresh tokens for access tokens and ID tokens. |
| job_execution_status | Background Job execution status (Ready/Running). Ready 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 |
| manager_approval_authoritative_source | Determines whether ServiceNow or One Identity Manager is the authoritative source for managers. |
| oneim_api_retrythreshold | The number of times the retry mechanism should be executed in case of failures. |
| oneim_mid_server_name | This parameter is used to specify the Mid Server Name, keep it blank if you don't want to use the mid server |
| oneim_request_ | <ul style="list-style-type: none"> If the valid from and valid until fields are not specified |

| Config Parameter Name | Config Parameter Description |
|---|--|
| validity_default | <p>during request creation, the system automatically applies this default validity.</p> <ul style="list-style-type: none"> • This ensures that requests always have a defined lifetime, even if no explicit dates are provided. • The parameter also accepts negative values: – If set to any negative value (For Eg. -1), and no validity dates are specified, the request is created without ValidFrom and ValidUntil values, resulting in unlimited validity. |
| oneim_rest_endpoint_url_api | REST API Endpoint to the One Identity Manager Application APIServer |
| 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_servicenow_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. |
| oneim_xml_max_childNode_search_count | Maximum number child nodes to search for a match in a xml document |
| perform_manager_approval | This parameter is used to determine ServiceNow request raised by the user should be approved by the manager or not. If set to true, request raised should be approved by the manager. If set to false, request raised need not be approved by the manager. (true false) |
| Request_approval_workflow_expire_in_days | The number of days post which the request workflow will expire and the requested service item will be aborted in One Identity manager if there is no activity on One Identity Manager for the requested service item |
| workflow_approval_timer | Retry interval in seconds to fetch the current status of the requested service item from One identity manager in request |

| Config Parameter Name | Config Parameter Description |
|-----------------------|---|
| interval_in_seconds | approval workflow of ServiceNow IMPORTANT: The default value is 3600 secs (60 minutes). Reducing this time limit could impact the performance of the ServiceNow instance. |

These configuration parameters are required for importing data into ServiceNow:

- oneim_rest_password
- oneim_rest_username
- oneim_rest_endpoint_url
- oneim_rest_endpoint_url_api

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

The following configuration parameters are required for authentication into ServiceNow only if using Single Sign-On (SSO):

- Identity Provider (IdP) Parameters
 - idp_auth_url
 - idp_client_id
 - idp_client_secret
 - idp_label
 - idp_redirect_uri
 - idp_tenant_id
 - idp_token_endpoint

Configuration Parameter

- auth_type – Specifies authentication type.
 - Options: Password, Both, OAuth.

Initial load from One Identity Manager to ServiceNow

Once the ServiceNow App is installed and configured, the One Identity Manager entities including **Person** and **Service Category** 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**.

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. 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 and their subordinates. All users synchronized into ServiceNow from One Identity Manager will be assigned to this role.

- **x_oni_oneim_addon.helpdesk** - This is the **One Identity Manager for Service Catalog** App helpdesk role. It is the responsibility of the SysAdmin to assign this role to appropriate users. Users with this Role can request service items for any user that has a matching identity record in One Identity Manager. These are detailed steps to configure helpdesk role –
 1. Assign **x_oni_oneim_addon.helpdesk** role to appropriate users.
 2. Assign **ServiceNowHelpDesk** role to the same user in one identity manager.
 3. Open Apiserver and login to Admin portal using system admin credential.
 4. Navigate to configuration and select "Web portal" in dropdown.
 5. Navigate to "Feature configuration(QER)".
 6. Append this query in "Identities for which request can be placed"

```
OR (EXISTS(SELECT 1 FROM PersonInAERole WHERE UID_Person = '%userid%'
AND UID_AERole = 'SCN-e47781323bd34e799e5ed9a2f4664f89') AND (uid_
person IN (SELECT uid_person FROM Person)))
```

Schedule job OneIdentity Manager user permissions required

Currently we support DialogUser authentication module and following are the minimum permissions required for the system user:

- IsServiceAccount should be enabled

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 |

OneIM_Requests

All the One identity requests are stored into this custom table. Any ServiceNow user can read data from this table.

| Column Name | Descriptions |
|-------------|--|
| Number | One Identity manager request number. |
| Stage | Stage of One Identity manager request |
| Opened | The time when the request was created. |

| | |
|-----------------------|--|
| Opened by | The serviceNow user who created the request. |
| State | Current state of request |
| Requested For | The user for which request is raised |
| Service category | Name of One identity manager service category |
| Service Item | Name of One identity manager service item |
| Service Item ID | Guide of One identity manager service item |
| SOD Result | Result of SOD check and validations |
| Reason | Reason for the request |
| Request Properties | This column stores the request properties of service item. |
| UID_ShoppingCartItem | Guid of One identity manager ShoppingCartItem |
| UID_shoppingCartOrder | Guid of One identity manager ShoppingCartOrder |
| Valid_From | The date from which request is valid |
| Valid_Until | The date until which request is valid |
| Short description | Short description for one identity manager request. |
| Active | This column specifies if the request is active or not. |

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 |

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

| | |
|------------|--|
| 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 (Deprecated)

This table is deprecated now. 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 (Deprecated)

This table is deprecated now. 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 (Deprecated)

This table is deprecated now. 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 | Name of the service category under which the Service Item is created in |

| | |
|----------|---|
| Category | One Identity Manager |
| User | Name of the user that has the resources |

Shopping Cart Order (Deprecated)

This table is deprecated now. 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. Deprecated tables were used in previous version of catalog application.

Login with SSO

Before raising a request in ServiceNow, users are redirected to login with One identity manager web portal credentials. But if customer wants to omit this login step then they need to setup SSO in their ServiceNow instance and one identity manager web portal. An example of SSO setup with Azure AD is given in the later part of this documentation.

Once ServiceNow and One identity manager both are setup with IDP, the following configuration parameters are important to provide:

- Auth_type – It should have value either “OAuth” or “Both”
- idp_auth_url
- idp_client-id
- idp_client_secret
- idp_redirect_uri – This should always be in this format - <https://<instance name>.service-now.com/sp?id=OneIMLoginPage>
- idp_tenant_id –
- idp_token_endpoint –

NOTE: Add following redirect URL in Identity provider.

<https://<instance name>.service-now.com/sp?id=OneIMLoginPage&source=others>

Setting Up Cryptographic Keys for the One Identity Manager Cryptographic Module

The **One Identity Manager Service Catalog Application** in ServiceNow requires cryptographic keys to secure data transmission and ensure application integrity. The cryptographic keys are used to securely encrypt the OneIM API server username and password entered by the user and store it temporarily in ServiceNow until the request is completed. Once the request is completed the secure credential is destroyed from ServiceNow. Before utilizing the application, clients must configure cryptographic keys for the cryptographic module named **oneim_cryptography_module**.

Prerequisites

- Access to the **Key Management** module in ServiceNow.
- Role required: **sn_kmf.cryptographic_manager**

To setup cryptographic key

1. Navigate to **Key Management > Cryptographic Modules > All**.
2. Locate and select **oneim_cryptography_module** to open its details page.
3. On the **Crypto Specifications** tab, find the row corresponding to the **Key Alias** entry.
| NOTE: If no key exists yet, the *Key Alias* field will be empty.
4. Click **Next** to navigate to the **Key Origin** tab under Crypto Specifications.
5. Enter a friendly name in the Key Alias field for easier identification.
6. Create the Key
 - a. Click **Next** to move to the **Key Creation** tab.
 - b. Select **Generate Key**.
 - i. Upon successful key generation, the **Cryptographic Module** form re-loads, displaying the updated Crypto Specification.
7. Verify the Generated Key
 - Navigate to the **Module Keys** tab to view and verify the generated keys.

Raising a request and approval workflow

One Identity Manager ServiceNow Application allows users that are assigned admin role/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 for themselves or their subordinates.

The resources are requested using the IT Shop from the ServiceNow catalog page. To raise a request for themselves, **Request For Self** can be selected. To raise a request for subordinates, **Request For others** can be selected. By default, users can raise a request for direct subordinate, but it can be extended to in-direct subordinates too. It requires configurational changes in APIServer (The required configurational changes are mentioned in later sections). See [Enabling Indirect Reportees for Request Creation](#).

The detailed procedure to request an IT Shop item 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. Click **Request Service Item**.

3. Users need to first login with **One identity manager web portal** credentials or SSO.
4. After login, users are redirected to the **User Picker** page, where they can raise requests for themselves or their subordinates.
5. Enter the Required details.
6. Validation is performed for each selected item before submission. If any selected item fails validation, the request cannot be submitted.
7. If the request passes all validations, proceed by clicking the submit button.

NOTE:

- Fetch specific service item for a user using key search: If a particular service item is not available in the picker or service category for the service item is not known, users can directly search for the item on a search bar, and can select the specific item.
- The request can be raised only from ServiceNow Service portal catalog page.

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

Once the request is approved from ServiceNow, the request is processed according to the approval policy applied on the requested service item in One Identity Manager. The request approval workflow of ServiceNow remains in the wait condition unless any activity (approve/reject) is performed from the One Identity Manager. The status of the request approval workflow of ServiceNow is updated accordingly.

User can change the number of times the request approval workflow executes using the max activity count property of workflow in ServiceNow.

Steps to change the max activity count

1. Navigate to the **Workflow->Workflow Editor** using the navigation bar of ServiceNow.
2. Click on the **Approval Workflow** for New Access Request.
3. Check out the workflow using the menu bar option.
4. Click on the properties.
5. Navigate to the Activities tab.
6. Change the max activity count value.
7. Publish the workflow using the menu bar option.

NOTE: If **Request_approval_workflow_expire_in_days** or max activity count condition is fulfilled, the ServiceNow request approval workflow is completed. The requested service item is aborted in the One Identity Manager if there is no activity on One Identity manager for the requested service item.

Adding more information to Approvers

The approval page in the global scope often lacks sufficient information related to specific One Identity Manager (OneIM) requests. To address this limitation, the OneIM Approval Record Widget can be used to provide comprehensive details about a OneIM request.

This widget ensures approvers have access to all relevant data, facilitating smoother and more informed decision-making.

Steps to Configure the OneIM Approval Record Widget

1. Navigate to Widgets in Service Portal
 - a. Go to **All > Service Portal > Widgets**.
2. Select the OneIM Approval Record Widget
 - a. Locate and select the **OneIM Approval Record Widget** from the list.
3. Clone the Widget in **Global Scope**
 - a. Clone the selected widget to ensure it operates within the global scope.
4. Modify the Approval Page
 - a. Go to **All > Service Portal > Pages**.
 - b. Find the **Approval Form** page and open it in the Designer.
5. Replace the Existing Widget
6. Replace the **Approval Record Widget** with the cloned version of the **OneIM Approval Record Widget**.

NOTE:

- The **OneIM Approval Record Widget** only modifies the content displayed for **One Identity Manager** requests.
- It does not impact the data or display of other requests on the approval page.
- This ensures more relevant and detailed information for **OneIM** requests while maintaining consistency for other request types.
- This customization adds a corresponding record in the Customer Updates [sys_update_xml] table. This table maintains the current version information for all objects that have been customized. During the upgrade process, any changes to objects with entries in this table are skipped. For more information, refer <https://www.servicenow.com/docs/bundle/yokohama-platform-administration/page/administer/upgrade-center/task/uc-revert-customization.html>.

Enabling Indirect Reportees for Request Creation

By default, the application allows users to raise requests for themselves and their direct reportees. However, this functionality can be extended to include indirect reportees as well.

Follow the steps below to enable this feature:

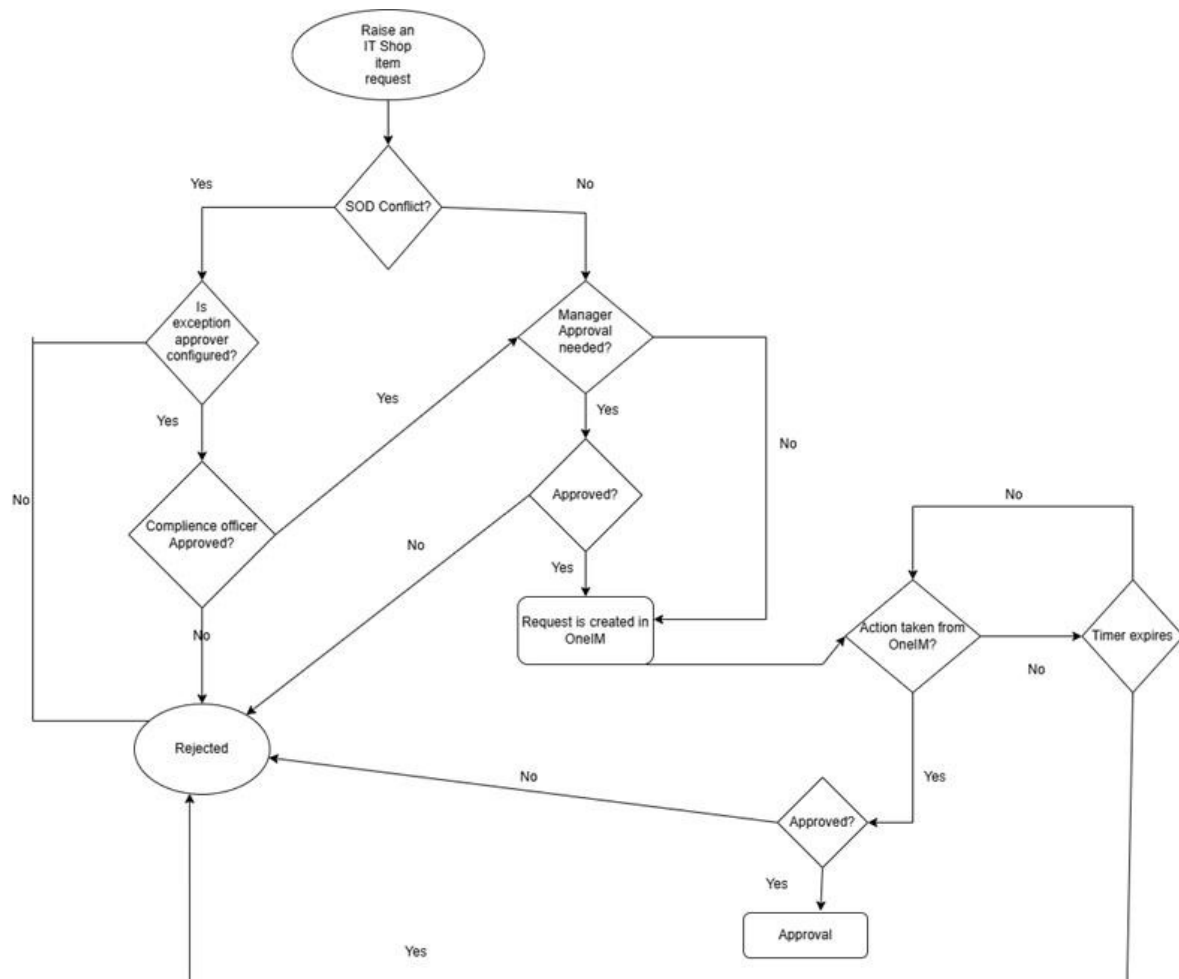
1. Open Apiserver and login to Admin portal using system admin credential.
2. Navigate to configuration and select "Web portal" in dropdown.
3. Navigate to "Feature configuration(QER)".
4. Append this query in "Identities for which request can be placed" –

```
OR (uid_person IN (SELECT uid_person FROM dbo.SCN_FGetRecursiveEmployees ('%userid%')))
```
5. Set "allow indirect reportees" to true.

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.

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:** A user can submit the request.
- **SOD Conflict for some of the requested items and exception approver has been configured in the One Identity Manager SOD Rule:** User can submit the request, but the request is routed to the compliance officer configured in ServiceNow (Configuration parameter: `compliance_officer`) post submission. 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*.

Customizing the Application

Clients can tailor the core functionality of the catalog application by leveraging Script Includes. While the default Script Includes containing core business logic are read-only, the application provides extended Script Includes for customization.

Clients can use these extended Script Includes to write their own business logic, enabling seam-less customization while preserving the integrity of the core application.

One Identity Manager Login Logging Customization Example

Scenario

Customers require enhanced logging during the application login process. This example demonstrates how to achieve this by leveraging inheritance and code overrides within the CommonHelper script.

Steps

1. Use extended CommonHelper Script
 - a. Open `commonHelperChild`.
 - b. Within `commonHelperChild`, define a new method named `loginToOneIMApiServer` that replicates the functionality of the existing method in `commonHelper`.

However, this new method should incorporate the desired logging functionality.

- c. Save the `commonHelperChild` script.
2. Clone OneIMLogin Widget
 - a. Create a copy of the `OneIMLoginPage` widget.
 - b. In the server-side script of the `OneIMLoginPage` widget, locate the reference to `commonHelper`.
 - c. Update the copied widget's script to reference the `commonHelperChild` script instead of `commonHelper`.
3. Update Application Page
 - a. Open the Service portal pages.
 - b. Navigate to `OneIMLoginPage` page and open the page in designer.
 - c. Replace the existing instance of the `OneIMLoginPage` widget with the cloned copy created in step 2.
4. Verify Functionality
 - a. Access the application through the Service Portal.
 - b. Attempt to log in to the application.
5. Review Logs
 - a. Navigate to the ServiceNow application log page.
 - b. You should now observe the additional login activity logs captured by the customized `loginToOneIMApiServer` method within `commonHelperChild`.

Create One Identity Manager employee from ServiceNow

One Identity Manager ServiceNow Application allows admin to create a new employee for One Identity Manager using ServiceNow "One Identity Manager for Person OnBoarding" feature. The detailed procedure to create an employee is explained below.

NOTE: Ensure that the data has been synced from One identity manager to ServiceNow tables.

To create an employee from ServiceNow Person OnBoarding Catalog page

1. From the ServiceNow instance portal navigate to the Catalog page.
2. Search for One Identity Manager for Person OnBoarding.
3. Enter the required details and click on submit.

Detailed explanation of the fields

Table 2: Fields

| Field Names | Description |
|-----------------------|---|
| FirstName | First name of the employee |
| LastName | Last name of the employee |
| Contact Email Address | The email address of the employee |
| Gender | Gender of the employee |
| Primary department | Department to which the employee is primary assigned |
| Primary Cost Center | Cost center to which the employee is primary assigned |
| Primary Location | Location to which the employee is primary assigned |
| Primary Business Role | Business role to which the employee is assigned |
| Person Manager | Select the Person Manager for the new employee |
| Person Sponsor | Select the Person Sponsor for the new employee. The sponsor is the ServiceNow user requesting for new employee |
| Date of Birth | This field will determine the date of birth of employee |
| Entry Date/Time | Date the employee started at the company. The Entry date is in user's configured timezone. The time will be converted into GMT format in the One Identity Manager |
| Employee Type | Employee type of the new Employee. This field is auto populated from the Configuration parameters "default_employee_type" |
| Remarks | Additional information about the Employee |

NOTE:

- Either one of the fields, "Person Manager" or "Person Sponsor" is mandatory. Person manager is given the preference if both are selected.
- Person Manager or Person Sponsor must have approver_role to approve the request.

- To view the Person OnBoarding form, the user must have x_oni_oneim_addon.admin role.

Enabling One Identity Manager's Shopping Cart in ServiceNow

One Identity Manager (ONEIM) provides a **Shopping Cart** feature that allows users to store requested items for later submission.

- Users who prefer to submit a request at a later time can use the Add to Cart button available on the final submission page.
- Items added to the shopping cart can be accessed directly using the following URL: `https://<instance-name>.service-now.com/sp?id=oneimcarts`

Adding ONEIM Shopping Cart to the Service Portal(sp) Header Menu

To make the ONEIM Shopping Cart accessible through the Service Portal header menu, follow these steps:

1. Ensure you are working in the **Global scope**.
2. Navigate to: **All > Service Portal > Menus**.
3. Open the record titled **SP Header Menu** (by clicking the **Updated** date column).
4. Click **New** to create a new menu item.
5. Complete the following details:
 - **Label:** OneIM Cart (or a preferred display name)
 - **Type:** Scripted List
 - **Condition:** `gs.isLoggedIn()`
6. In the **Script** field, add the following code:

```
// Limit header menu dropdown to 30 items  
var max = 30;  
var t = data;  
t.items = [];
```

```

t.count = 0;

// Record watchers to update dropdown counts dynamically

t.record_watchers = [];

t.record_watchers.push({

'table': 'x_oni_oneim_addon_oneim_requests',

'filter': 'submission_status=false^opened_by=' + gs.getUserID()

});

var z = new GlideRecord('x_oni_oneim_addon_oneim_requests');

z.addQuery("submission_status", false);

z.addQuery("opened_by", gs.getUserID());

z.orderByDesc('sys_updated_on');

z.setLimit(max);

z.query();

var link = {};

link.title = gs.getMessage('View all Items');

link.type = 'link';

link.href = '?id=oneimcarts';

link.items = [];

t.items.push(link);

while (z.next()) {

var a = {};

$sp.getRecordValues(a, z, 'short_description,sys_id,sys_updated_on');

a.title = z.getValue('service_item');

a.type = 'link';

a.href = '?id=oneimcart&table=' + z.getTable_name() + '&sys_id=' +

z.getUniqueValue();

```

```
t.items.push(a);  
t.count++;  
}  
if (t.count == 1)  
data.badgeHint = gs.getMessage('{0} item in cart', ['' + t.count]);  
else  
data.badgeHint = gs.getMessage('{0} items in cart', ['' + t.count]);
```

7. Click Submit to save the new menu item.

The **ONEIM Shopping Cart** will now appear in the Service Portal header.

- It will only be visible when the user has at least **one item** in their cart.

| **NOTE:** This cart is **different** from ServiceNow's native shopping cart.

One Identity Manager, Microsoft Entra ID, and ServiceNow MultiSSO Integration (OIDC)

This guide outlines the steps to configure Microsoft Entra ID (formerly Azure Active Directory or AAD) as an OpenID Connect (OIDC) provider, integrate it with both One Identity Manager (OneIM) and ServiceNow, and enable MultiSSO authentication.

Prerequisites

Ensure the following prerequisites are met:

- Azure AD Admin Access
- OneIM Designer and Web Designer access
- ServiceNow Admin Role
- Multiple Provider Single Sign-On Installer (com.snc.integration.sso.multi.installer) installed in ServiceNow
- Redirect URIs are registered properly in Azure and configurations in OneIM and ServiceNow

Register Application in Azure AD

1. Navigate to Microsoft Entra Admin Center (Azure Portal) → Microsoft Entra ID → App registrations → New registration.
2. Provide a name (e.g., OneIM_OIDC_Integration_App).
3. Choose Single tenant.

4. Under Redirect URIs (Web), add:
 - <https://<your-instance>.service-now.com/navpage.do>
 - <https://<your-instance>.service-now.com/sp>
 - <https://<your-instance>.service-now.com/sp?id=OneIMLoginPage&source=others>
5. Click Register.

Microsoft Graph API Permissions

Add the following Delegated permissions under Microsoft Graph:

- openid
- profile
- email
- User.Read
- offline_access

Steps to enable Graph API permissions

1. Go to **Azure Portal** → **App Registrations** → **Your App**.
2. Navigate to **API permissions**.
3. Click '+ **Add a permission**'.
4. Choose '**Microsoft Graph**' → '**Delegated permissions**'.
5. Select the above permissions and click '**Add permissions**'.
6. Click '**Grant admin consent**' for the tenant.

Create a Client Secret in Azure

1. Go to the **App** → **Certificates & secrets** → **New client secret**.
2. Add a description and expiry.
3. Click **Add**, then copy the secret immediately as you won't see it again.

Assign Users to the Azure AD Enterprise Application

1. Go to **Microsoft Entra ID** → **Enterprise Applications**.
2. Select the registered app.
3. Click **Users and groups** → **+ Add user/group**.
4. Select **users** → Click **Select** → **Assign**.

Gather Azure App Info

Collect the following for OneIM and ServiceNow configuration:

- Client ID
- Client Secret
- Tenant ID
- Discovery URL: <https://login.microsoftonline.com/<TenantID>/v2.0/.well-known/openid-configuration>
- Authorization URL (Auth URL):
<https://login.microsoftonline.com/<TenantID>/oauth2/v2.0/authorize>
- Token URL (Token Endpoint):
<https://login.microsoftonline.com/<TenantID>/oauth2/v2.0/token>

Configure OAuth in One Identity Manager

1. Open OneIM Designer.
2. Go to **Base Data** > **Security Settings** > **OAuth 2.0/OpenID Connect Configuration**.
3. Create a New Identity Provider.
4. Enter a Display Name.
5. Under Automatic configuration data discovery, enter the OpenID Connect metadata document URL and click Discover.
6. A message confirms the configuration has been loaded successfully. Click **OK**, then **Next**.

7. All configuration details (Login/Logout endpoints, Token endpoint, Issuer, Scope, UserInfo endpoint) should be pre-populated. Click **Next**.
8. On Configure certificates, the JSON Web Key endpoint should be pre-populated. Click **Next**.
9. On Search rule for user data:
 - a. Search value: email
 - b. Column to search: Default Email Address under the Person table.
 - c. Click **Next**.
10. On Create Application screen:
 - a. Click **New** and enter Display Name, Client ID, and Shared Secret under the General tab.
 - b. Go to Authentication tab:
 - i. Set Authentication Method to client_secret_post.
 - c. Click **Next**, then **Finish**.
11. Under Web server configurations:
 - a. Select your listed configurations (App Server and API Server).
 - b. Change Authentication Module to OAuth 2.0/OpenID Connect (role-based).
 - c. Set OAuth 2.0/OpenID Connect Application to the application created earlier.
12. Ensure both modules OAuth 2.0/OpenID Connect and OAuth 2.0/OpenID Connect (role-based) are enabled:
 - a. Go to Authentication Module under Security settings > Base Data.
 - b. Set Enabled = True in properties.
13. Commit changes to the database and click **Save**.

Enable MultiSSO Plugin in ServiceNow

1. Go to System Definition > Plugins.
2. Search for Integration - Multiple Provider Single Sign-On Installer and install. (com.snc.integration.sso.multi.installer)

Configure the OIDC Identity Provider in ServiceNow

1. Navigate to **MultiSSO > Identity Providers > New > OpenID Connect**.
2. Enter:
 - a. Name: AzureAD
 - b. Client ID & Secret: from Azure
 - c. Discovery Endpoint: (Discovery URL)
3. Click **Import** and then **Save**.
4. In the Identity Provider record, ensure the checkbox "**Show as login option**" is selected. This enables the "Login with Microsoft" button on the ServiceNow login page.

NOTE: Without this, users cannot see the SSO option and cannot use direct SSO links to authenticate with the integrated ONEIM Catalog app.

Testing the Integration

Testing in ServiceNow

1. Open a new incognito browser window.
2. Go to <https://<instance>.service-now.com/sp>.
3. On the Service Portal login page, click on the Azure SSO label (e.g., "Login with Microsoft").
4. You should be redirected to the Azure login screen.
5. After successful login, you will be redirected back to ServiceNow.

Testing in One Identity Manager

1. Open web portal: <https://<your-domain>/ApiServer>.
2. On the Web Portal page, select the Authentication as OAuth 2.0/OpenID Connect (role-based) and Click on Login.
3. You should be redirected to the Azure login screen.
4. After successful login, you will be redirected back to OneIM ApiServer.

Troubleshooting Tips

Table 3: Fields

| Issue | Solution |
|--------------------------------------|---|
| Redirect URI mismatch | Ensure exact URI is registered in Azure (case-sensitive). |
| Invalid client secret | Re-check expiry, regenerate if needed. |
| Discovery endpoint error | Confirm correct and reachable TenantID. |
| User not provisioned | Map required claims (email, UPN); ensure transform map or matching sys_user record exists. |
| Login button not visible | Enable "Show as login option" in the IdP record in ServiceNow. |
| Sign-in fails | Check Azure AD Sign-In Logs for detailed error messages. |
| No matching ServiceNow user | Ensure claim (e.g., email or upn) matches sys_user.email or user_name in ServiceNow. |
| No logs in ServiceNow | Enable glide.authenticate.multisso.debug in System Properties, check logs under SSO Logs. |
| Token validation fails | Select the Person Manager for the new employee. |
| Person Sponsor | Select the Person Sponsor for the new employee. The sponsor is the ServiceNow user requesting for new employee. |
| Date of Birth | This field will determine the date of birth of employee. |
| Scope-related errors (invalid_scope) | Validate scopes like openid, email, profile are included and allowed in Azure App config. |
| Claims not returned in token | Ensure claims like email, upn are configured under Azure AD > Token Configuration. |

| Issue | Solution |
|--|---|
| Unexpected logout behavior | Confirm logout endpoints are configured on both ServiceNow and Azure sides. |
| Azure AD Consent screen repeatedly appears | Grant admin consent or allow user consent to required Graph API permissions. |
| Token expiry too short | Adjust Access/ID/Refresh token lifetimes via Azure AD > Token Configuration. |

ServiceNow Mid Server Support

The Management, Instrumentation, and Discovery (MID) Server is a Java application that runs as a Windows service or UNIX daemon on a server in your local network. The ServiceNow® MID Server enables communication and the movement of data between a ServiceNow instance and external applications (Example: One Identity Manager App Server). MID Servers help you to control and secure how ServiceNow communicates with your organization's systems, especially those behind a firewall. This is optional and it can be used to communicate with the App Server, in case if direct communication to One Identity Manager App Server is not possible from ServiceNow.

You can setup a Mid Server by following the steps listed below.

Creating a MID Server Administrator User

1. Open the customer instance.
2. Navigate to **ALL | Organization | User**.
3. Create a new user and set the password, e.g.: Name "AdminMIDServer", First Name: Admin, Last Name: MidServer.
4. Set the role of the user as mid_server.

Creating a Service Account User

1. Navigate to Computer Management | Local Users and Groups | Users.
2. Create a new user account and set up the password.
3. Open Local Security Policy.
4. Navigate to **Local Security | User Rights Assignment | Log on as a Service**.
5. Add the newly created user.

Installing Mid Server in ServiceNow

1. Open the customer instance.
2. Navigate to **ALL | MID Server | Download**.
3. Download Windows (MSI) for Windows.
4. Create a Folder "MIDServer".
5. Copy Installation MSI in "MidServer" folder.
6. Run Installation.
7. Update values in "Configure Mid Server Connection Settings".
 - a. Authentication Type is Basic.
 - b. ServiceNow Instance URL is <Customer Instance URL>
 - c. ServiceNow MID Server UserName is < AdminMIDServer User Name>.
 - d. ServiceNow MID Server Password is < AdminMIDServer Password>.
8. Click **Test Your Connection**. It should display a successful message.
9. Click **Next**.
10. Update values in "Configure MID Server Service Settings."
 - a. MID Server Name, please update the name of the Mid Server, it can be any name of your choice. E.g. "OneIM_MIDServer".
 - b. Service Account Name, provide the name of the service account user.
 - c. Service Account Password, provide the password of the service account password.
11. Click **Validate MID Service Settings**. It should display the successful message.
12. Click **Next**.
13. Select the destination folder as C:/MIDServer.
14. Complete the installation.

Validating MID Server in ServiceNow

1. Open customer instance.
2. Navigate to **MID Server | Servers**.
3. Click Server Name that was added in previous steps.
4. Click Validate.

It should display "Status = Up" and "Validate = Validated".

Testing the MID Server

1. Open ServiceNow Instance.
2. Navigate to **All | Rest Messages**.
3. Click **Connection Test ServiceNow to One IDM**.
4. Scroll down and under HTTP Method click on "Logon to the Application Server".
5. Select the HTTP Request tab.
6. Enter the MID Server name in "Use MID Server" text box.
7. Enter username and password in Auth String under "HTTP Query Parameters"
8. Click **Test** and it should return the HTTP Status 200 with valid response.

How to use MID Server

A new configuration parameter called `oneim_mid_server_name` will be added to the list by the background job. The configuration parameter `oneim_mid_server_name` must be set to use the MID Server.

1. Navigate to **All | One Identity Manager for Service Catalog | Configuration Parameters**.
2. Set the configuration parameter with the MID Server Name.

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

The KMF module (x_oni_oneim_addon.oneim_cryptography_module) in ServiceNow could not be identified

To troubleshoot this issue, follow these steps:

1. **Test Login Page:** Check if the login page displays an error indicating "Invalid module name: x_oni_oneim_addon.oneim_cryptography_module".
2. **Verify Module Access Policies:** Ensure there are no auto-generated access policies associated with oneim_cryptography_module.
 - If any such policies exist, delete them and retest.

The error message "Error: The code you entered is not correct" displays during login

1. Access the API server and log in to the administrative portal using system administrator credentials.

2. Navigate to the following location: Configuration > API Server configuration > CAPTCHA login protection.
3. Disable CAPTCHA login protection by unchecking/deactivating the associated option.
4. Apply the changes to implement the updated configuration.

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