How to Sync Entities to an Import Set with Exalate for ServiceNow

Last Modified on 01/15/2026 5:47 pm EST

In ServiceNow, import sets act as an intermediate table between an external data source and a ServiceNow table. You can map fields between an import set and the main table with a transform map. Then you can sync the data from Exalate to an import set. This is useful if you don't want to immediately sync or import your data into a table.

This article shows how to sync data from Exalate to an import set.

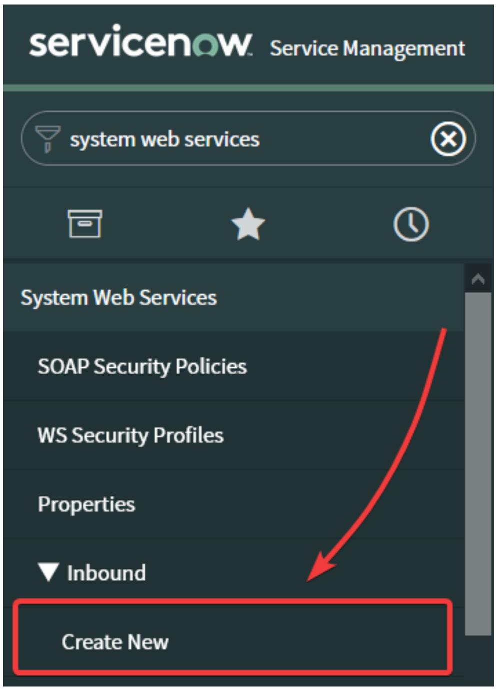
In this article

How to create a ServiceNow import set?

Exalate can sync data only to import sets that meet these requirements:

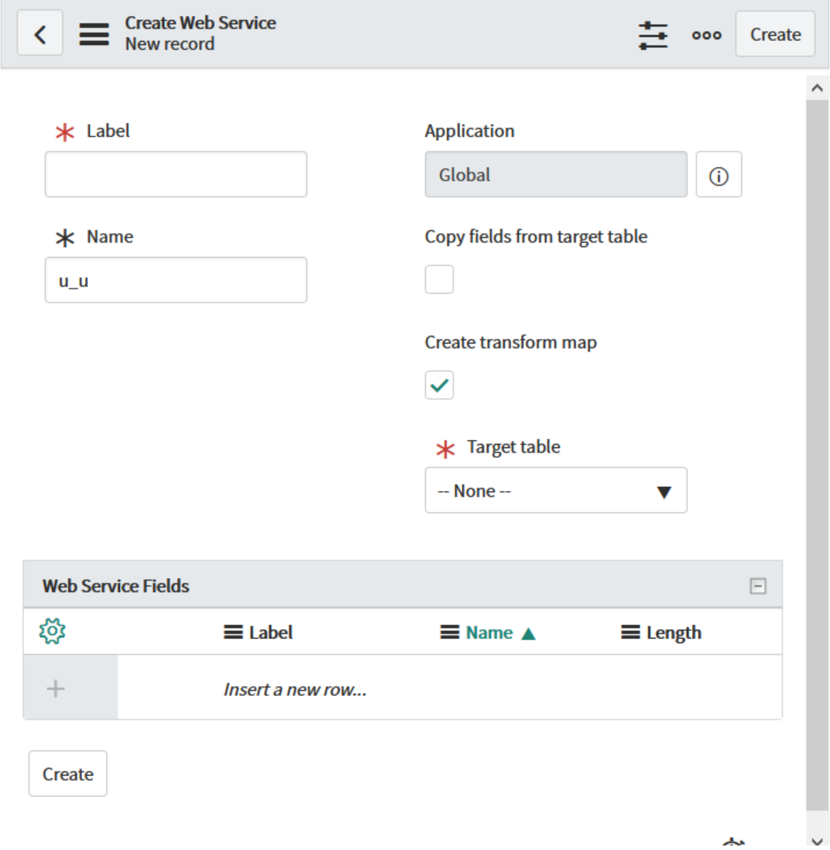
- **Coalesce on Empty Fields** is checked.
- There is a field in the import set that is mapped to the `sys_id` field in the ServiceNow table. It has the **Coalesce** option enabled.

1. Navigate to **System Web Services** -> **Create New**.



2. Fill in the fields for the import set.

To sync data successfully, you need to check **Create transform map** and fill in these fields: **Label**, **Name**, **Target table**.



*Label

Application: Global

*Name: u_u

Copy fields from target table □

Create transform map ☑

*Target table: -- None --

	Label	Name ▲	Length
+	*Insert a new row...*		

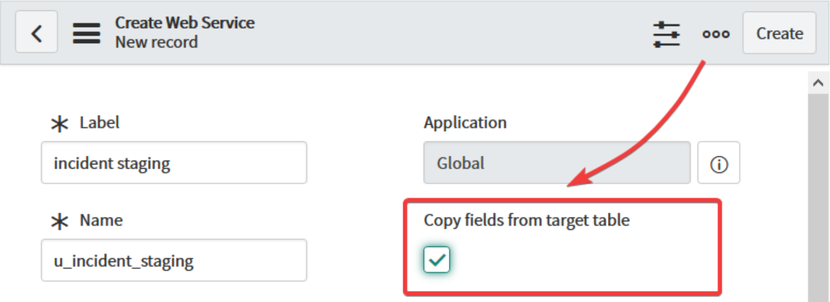
Field descriptions

Field	Description
Label	The name of the import set. Required field
Name	The internal name of the import set in ServiceNow. It is used by Exalate to sync data into the import set. Required field
Copy fields from target table	With this option, you can automatically create matching fields with the target table on the import set table
Create transform map	With this option, the import set is created with fields that match the target table
Target table	The ServiceNow table you want to map to the import set. The field appears only if **Create transform map** is checked

3. Fill in the **Web Service Fields**.

There are two ways to fill in the fields:

- Copy the identical fields from the target table by checking the **Copy fields from target table**.



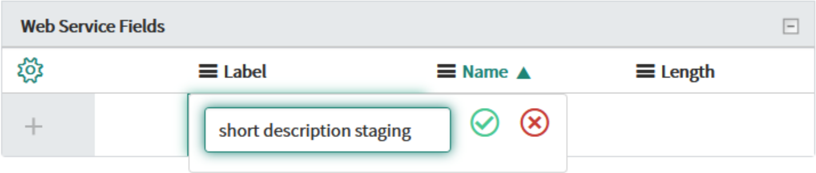
Label: incident staging

Name: u_incident_staging

Application: Global

Copy fields from target table ☑

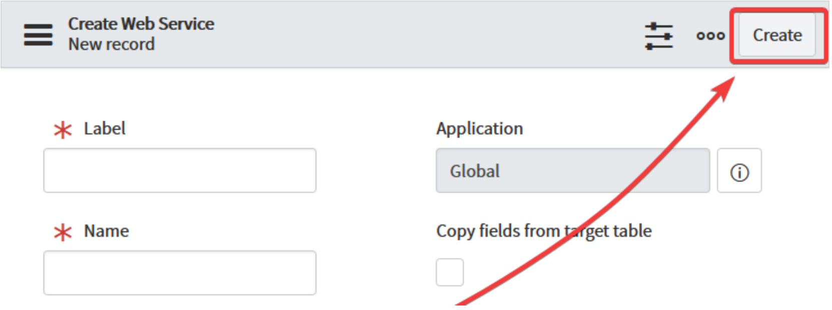
- Fill in each field manually in the **Web Service Fields** table.



Label: short description staging

4. Click **Create**.

If **Create transform map** is checked, you are redirected to the **Table Transform Map** screen to configure the transform map.



Label:

Name:

Application: Global

Copy fields from target table □

5. Fill in the fields for the transform map.

Make sure to check the **Create new record on empty coalesce fields** checkbox, so Exalate could sync entities to the import set. If the box is checked, a new record is created when all coalesce fields are empty. Otherwise, ServiceNow updates the existing record.

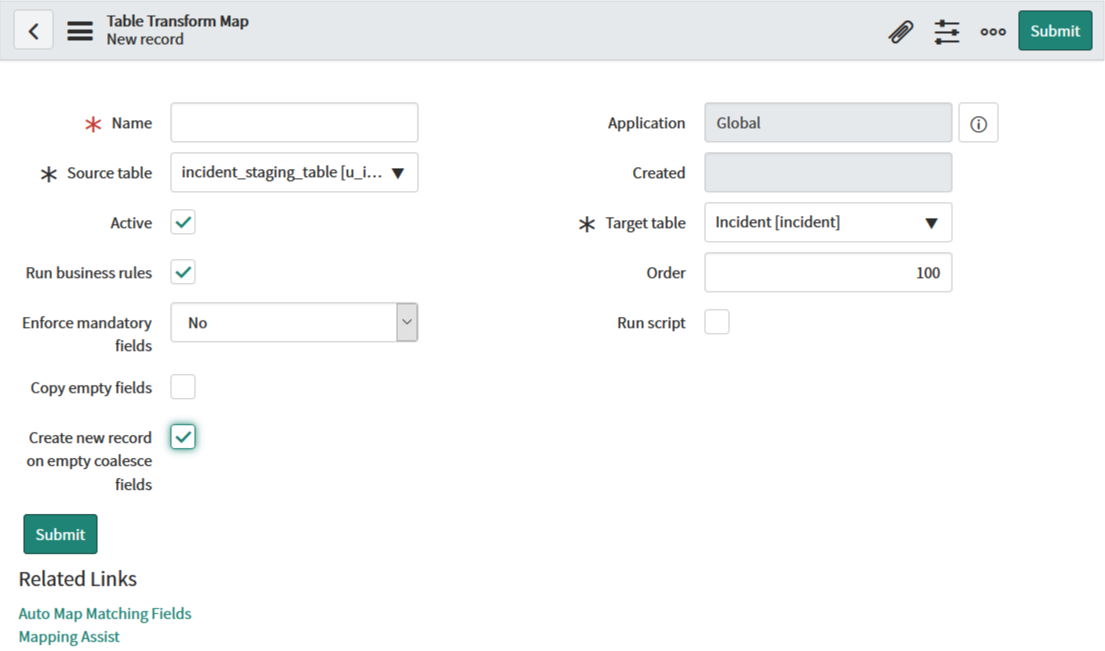


Table Transform Map
New record

Submit

* Name		Application	Global
* Source table	incident_staging_table [u_i…	Created	
Active	☑	* Target table	Incident [incident]
Run business rules	☑	Order	100
Enforce mandatory fields	No	Run script	□
Copy empty fields	□		
Create new record on empty coalesce fields	☑		

Submit

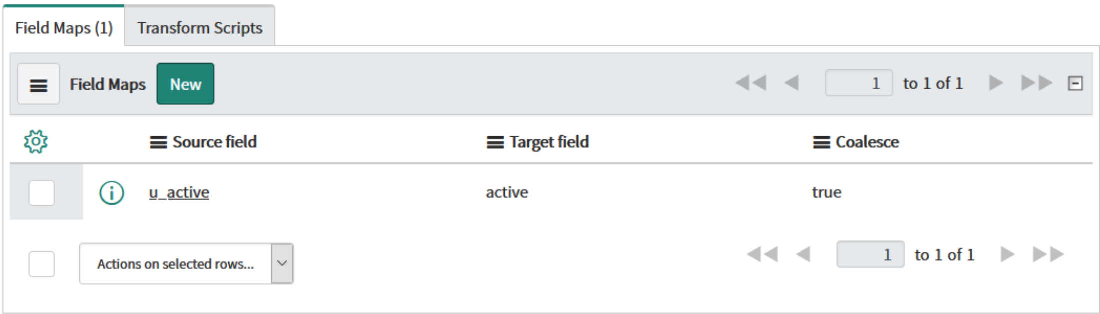
Related Links

Auto Map Matching Fields
Mapping Assist

6. Save the transform map.

You can save the transform map with either of these methods:

- Configure mapping automatically by clicking **Auto Map Matching Fields**.
- Click **Submit** and add the fields later in the **Field Maps** or **Transform Scripts** tab.



Field Maps (1) | Transform Scripts

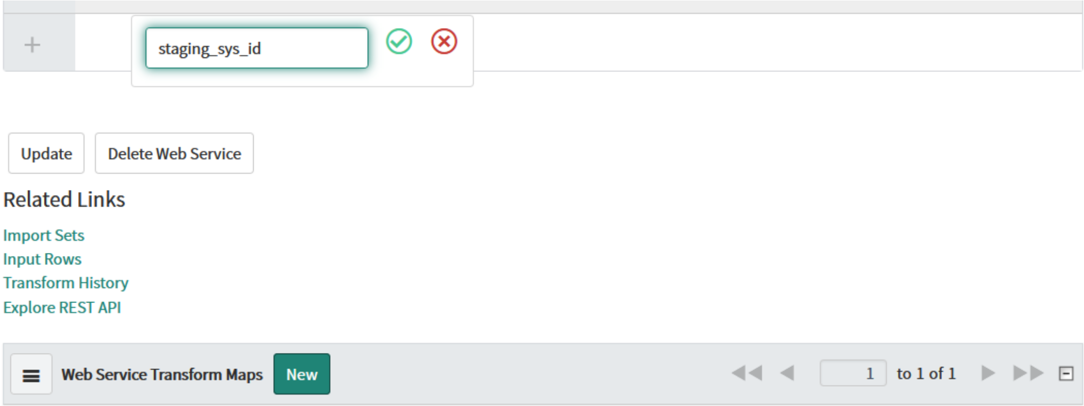
Field Maps New — 1 to 1 of 1

	Source field	Target field	Coalesce
□	u_active	active	true

□ Actions on selected rows... — 1 to 1 of 1

7. Create a field that is mapped to the sys_id of the ServiceNow table.

Enter the field name in the **Insert a new row** field and click ✓ . This field is used by Exalate to sync the `sys_id` .



+ staging_sys_id

Update | Delete Web Service

Related Links

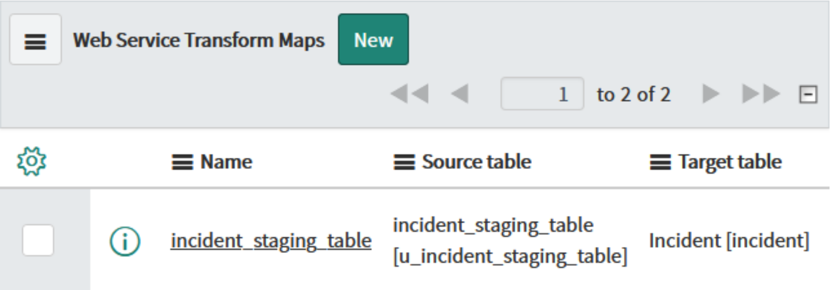
Import Sets
Input Rows
Transform History
Explore REST API

Web Service Transform Maps New — 1 to 1 of 1

8. Map the created field to the sys_id field.

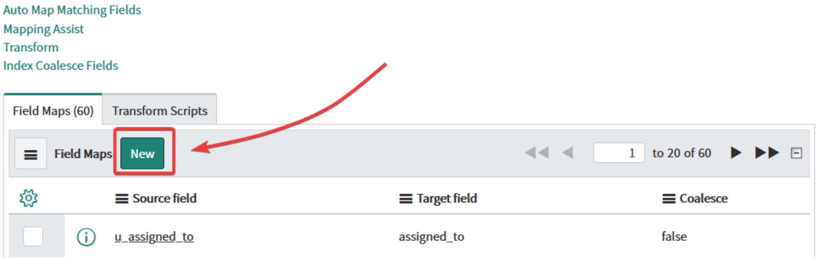
To map a field to the `sys_id` field:

1. Select the transform map in the **Web Service Transform Maps** table.



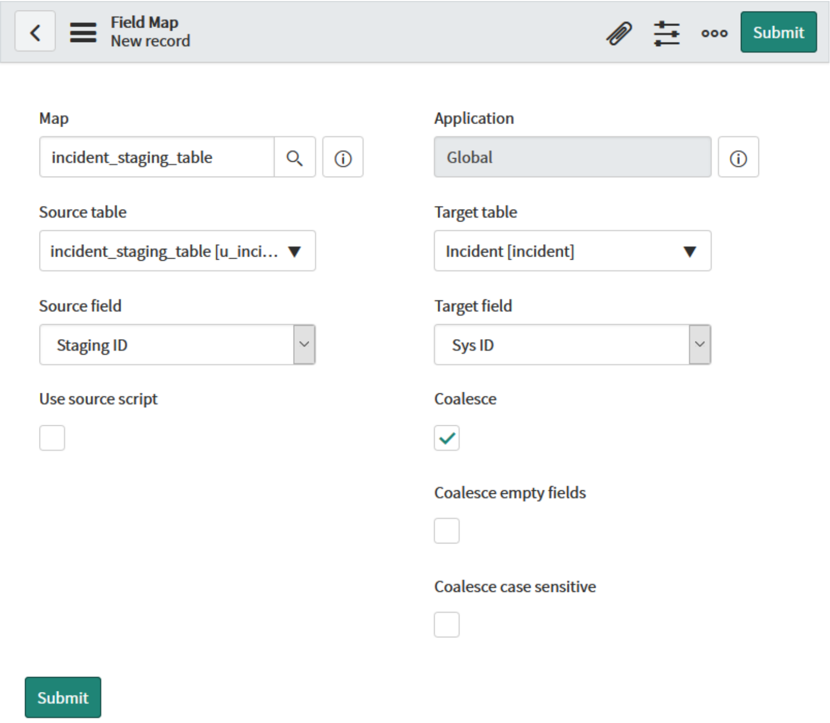
Name	Source table	Target table
incident_staging_table	incident_staging_table [u_incident_staging_table]	Incident [incident]

2. Select **New** in the **Field Maps** tab.



Source field	Target field	Coalesce
u_assigned_to	assigned_to	false

3. Select the created field in the **Source field** dropdown.



Map: incident_staging_table

Application: Global

Source table: incident_staging_table [u_inci...

Target table: Incident [incident]

Source field: Staging ID

Target field: Sys ID

Use source script: □

Coalesce: ☑

Coalesce empty fields: □

Coalesce case sensitive: □

4. Select **Sys ID** in the **Target Field** dropdown.
5. Check **Coalesce**. If this box is checked, this field gets updated if there are fields with the same target field and source field mapping.
6. click **Submit**.

Script example

The data is synced to import sets by adding the script to the incoming sync. The example below shows how to sync data to the import set mapped to the **Incident** table.

In order to sync data to an import set, you need to enter the name of the import set field. For example, to sync the short description, enter `entity.u_short_description` .

`u_short_description` is the name of the import set field mapped to the `short_description` field in the **Incident** table.

Incoming sync

```
if(firstSync){
 //Decide on the first sync, which entity you want to create based on the remote issue type
 if(replica.typeName == "Business Application"){
    entity.tableName = "cmdb_ci_business_app"
 }else{
    entity.tableName = "incident"
 }
}

if(entity.tableName == "incident") {
 entity.stagingTable = "incident_staging_table" //name of the staging table/import set
 entity.publicStagingComment = "u_comments" //field that will be used for your public comments
 entity.privateStagingComment = "u_work_notes" //field that will be used for your private comments
 entity.stagingSysId = "u_staging_sys_id" //name of the field you created to map to the production table's sys_id
    entity.u_short_description = replica.summary
    entity.u_description = replica.description
    entity.attachments += replica.addedAttachments
    entity.comments += replica.addedComments
    /*
    Jira Custom Field to ServiceNow Field
    Apply the value from a Jira custom field to the Resolution Notes
    This works for all other entity types as well
    entity.u_resolution_notes = replica.customFields."Jira CF Name".value
    */

    /*
    Status Synchronization
    Sync status according to the mapping [remote incident status: local incident status]
    If statuses are the same on both sides don't include them in the mapping
    def statusMapping = ["Open":"New", "To Do":"Open"]
    def remoteStatusName = replica.status.name
    entity.state = statusMapping[remoteStatusName] ?: remoteStatusName
   */
}

//any other entity can be synced using the table name and the entity variable
if(entity.tableName == "cmdb_ci_business_app") {
    entity.short_description = replica.summary
    entity.description = replica.description
}
```

Script variables

Variable	**Description**
`incident_staging_table`	Internal name of the import set

Variable	Description
`u_comments`	Internal name of the field that is used for public comments
`u_work_notes`	Internal name of the field that is used for private comments
`u_staging_sys_id`	Internal name of the field you created to map to the production table's `sys_id` .

You can check the internal names of the fields in the **Name** column of the **Web Service Fields** table.

> Have more questions? Ask the community