

How to Sync Any Entity in Exalate for Salesforce

Last Modified on 01/30/2026 10:25 am EST

Using Exalate you can sync any entity from Salesforce. This article describes examples of how to sync various objects. Scripts below use objects and field names from the Salesforce database. You can [find out the field and object names of any entity in Salesforce](#).

Outgoing Sync

On the outgoing sync, you define which objects you want to send. Use the variable **entity.entityType** to define the entity you are executing the script for. With this script, you can sync fields from the **Case** and **Task** entities. This script shows how to sync the following fields from the Case object:

- Id
- Subject
- Description
- Comments
- Attachments
- Status

This script shows how to sync the following fields from the Task object:

- Id
- Subject
- Description
- Status

```
if(entity.entityType == "Case") {  
    replica.key      = entity.Id  
    replica.summary   = entity.Subject  
    replica.description = entity.Description  
    replica.comments  = entity.comments  
    replica.attachments = entity.attachments  
    replica.Status    = entity.Status  
}  
/*  
Use a field's internal name to send its value  
Example: Call Type -> CallType  
This works for all other entity types as well  
replica.CallType = entity.CallType  
*/  
}  
//any other entity can be synced using the object name and the entity variable  
if(entity.entityType == "Task") {  
    replica.key      = entity.Id  
    replica.summary   = entity.Subject  
    replica.description = entity.Description  
    replica.Status    = entity.Status  
}
```

Script Variables

entity.entityType

A variable used to know what object to sync. Assign the name of the Salesforce object name to this variable. This example shows how to sync data from two objects: `case` and `task`. You can sync any entity within the `if` condition. For example, with the `replica.key = entity.Id` line, you sync the `key` field from the `case` object. You can add more fields, like `description` or `comments`.

Incoming Sync

This script shows how to set up incoming sync for Salesforce entities. This example script shows how to sync issues between Salesforce and Jira.

On the first synchronization, it's important to define what Salesforce entities you want to create after receiving data from Jira. This is done with the `if(firstSync)` variable. In this example, **Case** is used as the default entity, where all data is synced. The fields from the Task object are synced in the Task entity.

```
if(firstSync){  
    //Decide on the first sync, which entity you want to create based on the remote issue type  
    if(replica.type.name == "Task"){  
        entity.entityType = "Task"  
    }else{  
        entity.entityType = "Case"  
    }  
    if(entity.entityType == "Case") {  
        entity.Subject = replica.summary  
        entity.Description = replica.description  
        entity.comments = commentHelper.mergeComments(entity, replica)  
        entity.attachments = attachmentHelper.mergeAttachments(entity, replica)  
    /*  
     * Jira Custom Field to Salesforce Field  
     * Apply the value from a Jira custom field to the Business Hours  
     * This works for all other entity types as well  
     * entity.BusinessHoursId = replica.customFields."Jira CF Name".value  
    */  
    /*  
     * Status Synchronization  
     * Sync status according to the mapping [remote issue status: local case status]  
     * If statuses are the same on both sides don't include them in the mapping  
     * def statusMapping = ["Open":"New", "To Do":"Escalated"]  
     * def remoteStatusName = replica.status.name  
     * entity.Status = statusMapping[remoteStatusName] ?: remoteStatusName  
    */  
    }  
    //any other entity can be synced using the object name and the entity variable  
    if(entity.entityType == "Task") {  
        entity.Subject = replica.summary  
        entity.Description = replica.description  
    }  
}
```

Script variables

if(firstSync)

A condition where that defines the objects used to store incoming data in Salesforce. With this script, you can store data in the `Case` and `Task` objects.

entity.entityType

A variable used to know what object you can sync. Assign the name of the Salesforce object to this

variable. In this example, you can sync data into two objects: `Case` and `Task`. You can sync any entity within the `if` condition. For example, with the `entity.Subject = replica.summary` line, you sync the `summary` field from Jira issues into the `Subject` field from `Case` object. You can add other fields, like `Description`.

```
def statusMapping
```

Sets mapping for statuses according to the following template `[remote issue status: local case status]`. If you set the mapping from the example in Salesforce, the `New` status in Salesforce is displayed as `Open` in Jira.

Creating a Trigger

If you want an entity to be synced automatically when it matches a Salesforce query, you need to create a trigger for it in the [Triggers](#) tab. To create a trigger for any entity (object) on your Salesforce ticket, start typing the object name on the entity select field:

Note: If there is only one entity type available, the select dropdown menu is disabled.

Have more questions? [Ask the community](#)

Video Tutorial (Only for the Classic Exalate experience)

ON THIS PAGE

[Product](#)
[Outgoing Sync](#)
[About Us](#)

[Referencing Story](#)

[Glossary](#)
[Creating a Trigger](#)
[API Reference](#)

[Video Tutorial \(Only for the Classic Exalate experience\)](#)

[Security](#)
[Pricing and Licensing](#)

Resources

[Subscribe for a weekly Exalate hack](#)

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

Still need help?

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)