How to Configure Sync to Multiple Projects in Azure DevOps Server & Service

Last Modified on 10/21/2025 2:29 am EDT

Starting from **Exalate for Azure DevOps version 5.4.0** , you can configure sync to multiple projects in Azure DevOps while using only one Script connection.

This feature saves you lots of time when configuring initial synchronization logic. It also makes maintenance much easier in the future.

You can use this functionality to connect your Azure DevOps with multiple entities in any work management system supported by Exalate, or another ADO instance.

> **Note**: If you plan to convert an **existing connection,** please reach out to our support department to smooth your transition.

There are several ways to perform the multi-project sync.

In one case, it can be performed between 2 instances that have multiple project support like Azure DevOps and Jira, Salesforce, or Github.

In the following example, you can see how to sync issues **from multiple Jira Cloud projects to multiple Azure DevOps projects.**

Azure DevOps Incoming Sync Rules

```
if(firstSync){
  // Set type name from source entity, if not found set a default
  if (replica.project.key == "Jira-A"){
     workItem.projectKey  =  "Azure-A"
     workItem.typeName = nodeHelper.getIssueType(replica.type?.name)?.name ?: "Task";
  } else if(replica.project.key == "Jira-B") {
     workItem.projectKey  =  "Azure-B"
     workItem.typeName = nodeHelper.getIssueType(replica.type?.name)?.name ?: "Task";
  } else {
    workItem.projectKey  =  "Azure-C"
    workItem.typeName = nodeHelper.getIssueType(replica.type?.name)?.name ?: "Task";
  }
}
```

In the example above, issues from Jira Cloud project "Jira-A" are synced into Azure DevOps project "Azure-A", and issues from Jira Cloud project "Jira-B" are synced into Azure DevOps project "Azure-B".

Any other Jira Cloud project issue is synced into Azure DevOps project "Azure-C".

Another way to configure multi-project sync is to send some specific entity types from the source side into Azure DevOps multiple projects. In this case, the source side does not need to support

multiple projects.

In the following example, you can see how to sync **different entities types from ServiceNow to multiple Azure DevOps projects**:

ServiceNow Outgoing Sync Rules

```
if(entity.tableName == "incident") {
    replica.key           = entity.key
    replica.summary       = entity.short_description
    replica.description   = entity.description
    replica.attachments   = entity.attachments
    replica.comments      = entity.comments
    replica.state         = entity.state
    replica.tableName     = entity.tableName
}
if(entity.tableName == "problem") {
    replica.key           = entity.key
    replica.summary       = entity.short_description
    replica.description   = entity.description
    replica.attachments   = entity.attachments
    replica.comments      = entity.comments
    replica.state         = entity.state
    replica.tableName     = entity.tableName
}

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

Azure DevOps Incoming Sync Rules

```
if(firstSync){

  if (replica.tableName == "incident"){
     workItem.projectKey  =  "Azure-A"
     workItem.typeName = nodeHelper.getIssueType(replica.type?.name)?.name ?: "Task";
  } else if(replica.tableName == "problem") {
     workItem.projectKey  =  "Azure-B"
     workItem.typeName = nodeHelper.getIssueType(replica.type?.name)?.name ?: "Task";
  } else {
    workItem.projectKey  =  "Azure-C"
    workItem.typeName = nodeHelper.getIssueType(replica.type?.name)?.name ?: "Task";
  }
}
```

In the example above, the incidents from ServiceNow are synced into Azure DevOps project "Azure-A", problems are synced into project "Azure-B", and any other entity type is synced to the project "Azure-C".