

How to Sync Insight Custom Fields

Last Modified on 02/11/2026 9:29 am EST

This article shows examples of syncing Insight custom fields.

Exalate can sync custom fields created in [Insight](#).

Syncing an Insight Custom Field

In this example, we sync an `Assets` custom field.

Source Side

Outgoing sync

```
// SETTINGS
final def insightCustomFieldName = "Assets"
// END SETTINGS

replica.customKeys.
"My Custom Field values as Strings" = issue.customFields[insightCustomFieldName]?.value?.collect{
  v ->
    def cfm = com.atlassian.jira.component.ComponentAccessor.getCustomFieldManager()
    def cf = cfm.getCustomFieldObject(issue.customFields[insightCustomFieldName].id)
    def cft = cf.getCustomFieldType()
    def vStr = cft.getStringFromSingularObject(v)
    vStr
  }
}
```

Destination Side

Incoming sync

```
// SETTINGS
final def insightCustomFieldName = "Assets"
// END SETTINGS

issue.customFields[insightCustomFieldName].value = replica.customKeys.
"My Custom Field values as Strings".collect {
  String vStr ->
    def cfm = com.atlassian.jira.component.ComponentAccessor.getCustomFieldManager()
    def cf = cfm.getCustomFieldObject(issue.customFields[insightCustomFieldName].id)
    def cft = cf.getCustomFieldType()
    def v = cft.getSingularObjectFromString(vStr)
    v
  }
}
```

Syncing Multiple Insight Custom Fields

In this example, we show how to synchronize 5 Insight custom fields:

- `CMDB Customer`
- `CMDB SLAContract`
- `CMDB Simulator`

- CMDB Site
- CMDB User

Source Side

Outgoing sync

```
// Insight Custom Fields to replicate
final def insightCMDBCustomer = "CMDB Customer"
final def insightCMDBSLAContract = "CMDB SLAContract"
final def insightCMDBSimulator = "CMDB Simulator"
final def insightCMDBSite = "CMDB Site"
final def insightCMDBUser = "CMDB User"

final def insightFields = [insightCMDBCustomer, insightCMDBSLAContract, insightCMDBSimulator, insightCMDBSite, insightCMDBUser]

// General Data
replica.key = issue.key
replica.type = issue.type
replica.assignee = issue.assignee
replica.reporter = issue.reporter
replica.summary = issue.summary
replica.description = issue.description
replica.labels = issue.labels
replica.comments = issue.comments
replica.resolution = issue.resolution
replica.status = issue.status
replica.parentId = issue.parentId
replica.priority = issue.priority
replica.attachments = issue.attachments
replica.project = issue.project
replica.components = issue.components

//log.warn("Insight: ${issue.customFields[insightCMDBCustomer]?.value?.getValue()}")
insightFields.forEach() { insf ->
  //log.warn(insf)
  replica.customKeys."${insf}" = issue.customFields[insf]?.value?.collect { v ->
    //log.warn("Insight: Field "${insf}", Value "${v?.getName()}"")
    v?.getName()
  }
}
```

Destination Side

Incoming sync

```

if(firstSync){
  // If it's the first sync for an issue (local issue does not exist yet)
  // Set project key from source issue, if not found set a default
  issue.projectKey = "L3HSUPPORTM"
  // Set type name from source issue, if not found set a default
  //issue.typeName = "Service Request"
  issue.typeName = nodeHelper.getIssueType(replica.type?.name, issue.projectKey)?.name ?: "Service Request"
}
issue.summary = replica.summary
issue.description = replica.description
issue.labels = replica.labels

// Comments
//issue.comments = commentHelper.mergeComments(issue, replica)
replica.addedComments.each { it.executor = nodeHelper.getUserByUsername(it.author?.username) }
replica.changedComments.each { it.executor = nodeHelper.getUserByUsername(it.updateAuthor?.username) }
issue.comments = commentHelper.mergeComments(issue, replica, { it })
// issue.comments = commentHelper.mergeComments(issue, replica)

// Attachments
issue.attachments = attachmentHelper.mergeAttachments(issue, replica)

// Components
issue.components = replica.components.collect{
  nodeHelper.createComponent(
    issue,
    it.name,
    it.description,
    it.leadKey,
    it.assigneeType.name()
  )
}

// Insight Custom Fields to replicate
final def insightCMDBCustomer = "CMDB Customer"
final def insightCMDBSLAContract = "CMDB SLAContract"
final def insightCMDBSimulator = "CMDB Simulator"
final def insightCMDBSite = "CMDB Site"
final def insightCMDBUser = "CMDB User"

final def insightFields = [insightCMDBCustomer, insightCMDBSLAContract, insightCMDBSimulator, insightCMDBSite, insightCMDBUser]

final def sdCMDBDetails = "SD CMDB Details"

//log.warn("Insight: ${replica.customKeys."${insightCMDBCustomer}"?.value.join("\n")} """)

issue.customFields[sdCMDBDetails].value = ""
||CMDB Customer |${replica.customKeys."${insightCMDBCustomer}"?.value?.join("\n") ?: " " }|
||CMDB SLAContract|${replica.customKeys."${insightCMDBSLAContract}"?.value?.join("\n") ?: " " }|
||CMDB Simulator |${replica.customKeys."${insightCMDBSimulator}"?.value?.join("\n") ?: " " }|
||CMDB Site |${replica.customKeys."${insightCMDBSite}"?.value?.join("\n") ?: " " }|
||CMDB User |${replica.customKeys."${insightCMDBUser}"?.value?.join("\n") ?: " " }|

""

```

Product

[About Us](#)

[Release History](#)

ON THIS PAGE

[Glossary](#)

[Syncing an Insight Custom Field](#)

[API Reference](#)

[Security](#)

[Syncing Multiple Insight Custom Fields](#)

[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](#)