

# Organization Sync is Broken

Last Modified on 04/01/2026 7:03 am EDT

## Problem

When syncing the Organization field between two Service Desk projects in Jira on-premise, the following error occurs:

### Error detail:

```
Exalate has problems while trying to create an issue in this Jira.
Details: Error occurred while creating issue. This could be due to a plugin being incompatible with this version of JIRA.
For more details please consult the logs, and see: http://confluence.atlassian.com/x/3McB - com.exalate.basic.domain.
hubobject.v1.BasicHubOrganization cannot be cast to com.atlassian.servicedesk.api.organization.CustomerOrganization
```

## Cause

The issue is caused by the fact that Exalate does not convert its own class to the Atlassian class.

## Solution

To sync the Organization names between the Service Desk projects, you can add the following Sync rules to the Outgoing and Incoming sync:

### Outgoing Sync Rules

```
def getCustomerOrganizations = { _proxyUser, serviceDeskId ->
    com.atlassian.servicedesk.api.organization.OrganizationService orgServ = com.atlassian.jira.component.Component
    tAccessor
        .getOSGiComponentInstanceOfType(com.atlassian.servicedesk.api.organization.OrganizationService.class);

    com.atlassian.servicedesk.api.organization.OrganizationsQuery organizationsQuery =
        orgServ.newOrganizationsQueryBuilder()
            .pagedRequest(com.atlassian.servicedesk.api.util.paging.SimplePagedRequest.paged(0, 50))
            .serviceDeskId(serviceDeskId)
            .build();

    Object either = orgServ
        .getOrganizations(_proxyUser, organizationsQuery)
    com.atlassian.servicedesk.api.util.paging.PagedResponse<com.atlassian.servicedesk.api.organization.CustomerOr
    ganization> orgPage = either
        .right()
        .getOrThrow(new com.google.common.base.Supplier<com.exalate.api.exception.IssueTrackerException>() {
    public com.exalate.api.exception.IssueTrackerException get() { return new com.exalate.api.exception.IssueTrackerEx
    ception("Something failed!"); });

    List<com.atlassian.servicedesk.api.organization.CustomerOrganization> results = com.google.common.collect.List
    s.newArrayList();
    results.addAll(orgPage.getResults());

    int limitToKeepTrack = 0;
    while (orgPage.hasNextPage()) {
        limitToKeepTrack += 50;
```

```

        limitToKeepTrack += 50;
        organizationsQuery = orgServ.newOrganizationsQueryBuilder()
            .pagedRequest(com.atlassian.servicedesk.api.util.paging.SimplePagedRequest.paged(limitToKeepTrack, limitToKeepTrack + 50))
            .serviceDeskId(serviceDeskId)
            .build();

        orgPage = orgServ.getOrganizations(_proxyUser, organizationsQuery);

        results.addAll(orgPage.getResults())
    }

    return results;
}

def getLocalOrgs = { jIssue ->
    def serviceDeskManagerAdapter = com.atlassian.jira.component.ComponentAccessor.getOSGiComponentInstanceOfType(com.exalate.compatibility.api.IServiceDeskManagerAdapter.class)
    Integer serviceDeskId = serviceDeskManagerAdapter.getServiceDeskIdForProject(jIssue.getProjectObject());
    def proxyUser = com.atlassian.jira.component.ComponentAccessor.getOSGiComponentInstanceOfType(com.exalate.api.node.INodeService.class)
        .proxyUser
    if (serviceDeskId == null) {
        log.error("Service desk is not available for project with id " + jIssue.getProjectId());
        return null;
    }

    return getCustomerOrganizations(proxyUser, serviceDeskId);
}

def toHubOrg = { com.atlassian.servicedesk.api.organization.CustomerOrganization org ->
    new com.exalate.basic.domain.hubobject.v1.BasicHubOrganization(org.name, [])
}

def getOrgs = {
    def customFieldManager = com.atlassian.jira.component.ComponentAccessor.customFieldManager
    def cf = customFieldManager.getCustomFieldObject(issue.customFields.Organizations.id as Long)
    def issueManager = com.atlassian.jira.component.ComponentAccessor.issueManager
    def jIssue = issueManager.getIssueObject(issue.id as Long)
    def localOrgs = getLocalOrgs(jIssue)
    def orgs = localOrgs.collect(toHubOrg)
    orgs
}

replica.customFields.Organizations = issue.customFields.Organizations
replica.customFields.Organizations.value = getOrgs()

```

## Incoming Sync Rules

```

def proxyUser = com.atlassian.jira.component.ComponentAccessor.getOSGiComponentInstanceOfType(com.exalate.api.node.INodeService.class)
    .proxyUser
def toCustomerOrganizations = { _proxyUser, serviceDeskId, List<String> valueOrgNames ->
    com.atlassian.servicedesk.api.organization.OrganizationService orgServ = com.atlassian.jira.component.ComponentAccessor
        .getOSGiComponentInstanceOfType(com.atlassian.servicedesk.api.organization.OrganizationService.class);

    com.atlassian.servicedesk.api.organization.OrganizationsQuery organizationsQuery =
        orgServ.newOrganizationsQueryBuilder()
            .pagedRequest(com.atlassian.servicedesk.api.util.paging.SimplePagedRequest.paged(0, 50))
            .serviceDeskId(serviceDeskId)
            .build();

    Object either = orgServ
        .getOrganizations(proxyUser, organizationsQuery)

```

```

    com.atlassian.servicedesk.api.util.paging.PagedResponse<com.atlassian.servicedesk.api.organization.CustomerOr
organization> orgPage = either
        .right()
        .getOrThrow(new com.google.common.base.Supplier<com.exalate.api.exception.IssueTrackerException>() {
public com.exalate.api.exception.IssueTrackerException get() { return new com.exalate.api.exception.IssueTrackerEx
ception("Something failed!")});});

    List<com.atlassian.servicedesk.api.organization.CustomerOrganization> results = com.google.common.collect.Lists.newArrayList();
    for (com.atlassian.servicedesk.api.organization.CustomerOrganization org : orgPage.getResults()) {
        if (valueOrgNames.contains(org.getName())) {
            results.add(org);
        }
    }

    int limitToKeepTrack = 0;
    while (orgPage.hasNextPage()) {
        limitToKeepTrack += 50;
        organizationsQuery = orgServ.newOrganizationsQueryBuilder()
            .pagedRequest(com.atlassian.servicedesk.api.util.paging.SimplePagedRequest.paged(limitToKeepTrack, lim
itToKeepTrack + 50))
            .serviceDeskId(serviceDeskId)
            .build();

        orgPage = orgServ.getOrganizations(proxyUser, organizationsQuery);

        for (com.atlassian.servicedesk.api.organization.CustomerOrganization org : orgPage.getResults()) {
            if (valueOrgNames.contains(org.getName())) {
                results.add(org);
            }
        }
    }

    return results;
}
def getLocalOrganizations = { jIssue, value ->
    def serviceDeskManagerAdapter = com.atlassian.jira.component.ComponentAccessor.getOSGiComponentInstance
OfType(com.exalate.compatibility.api.IServiceDeskManagerAdapter.class)
    Integer serviceDeskId = serviceDeskManagerAdapter.getServiceDeskIdForProject(jIssue.getProjectObject());
    if (serviceDeskId == null) {
        log.error("Service desk in not available for project with id " + jIssue.getProjectId());
        return null;
    }

    List<String> valueOrgNames = com.google.common.collect.Lists.newArrayList();
    if (value instanceof List && ((List) value).size() > 0 && ((List) value).get(0) instanceof com.atlassian.servicedesk.api
.organization.CustomerOrganization) {
        return value;
    }

    if(value instanceof com.atlassian.servicedesk.api.organization.CustomerOrganization){
        return com.google.common.collect.Lists.newArrayList(value);
    }

    if (value instanceof List && ((List) value).size() > 0 && ((List) value).get(0) instanceof String) {
        valueOrgNames = (List<String>)value;
    }

    if (value instanceof List && ((List) value).size() > 0 && ((List) value).get(0) instanceof com.exalate.api.domain.hubo
bject.IHubOrganization) {
        List<com.exalate.api.domain.hubobject.IHubOrganization> organizations = (List<com.exalate.api.domain.hubob
ject.IHubOrganization>) value;
        valueOrgNames = com.google.common.collect.Lists.newArrayList(com.google.common.collect.Iterables.transfor
m(organizations, new com.google.common.base.Function<com.exalate.api.domain.hubobject.IHubOrganization, Strin

```

```

g>() {
    public String apply(com.exalate.api.domain.hubobject.IHubOrganization org) {
        return org.getOrganizationName();
    }
});
}

if(value instanceof String){
    valueOrgNames = com.google.common.collect.Lists.newArrayList(value.toString());
}

if(value instanceof com.exalate.api.domain.hubobject.IHubOrganization){
    valueOrgNames = com.google.common.collect.Lists.newArrayList(((com.exalate.api.domain.hubobject.IHubOrga
nization)value).getOrganizationName());
}
return toCustomerOrganizations(proxyUser, serviceDeskId, valueOrgNames);
}
def setOrganizations = { remoteCf ->
    def remoteOrgs = remoteCf.value
    def customFieldManager = com.atlassian.jira.component.ComponentAccessor.customFieldManager
    def cf = customFieldManager.getCustomFieldObject(issue.customFields.Organizations.id as Long)
    def issueManager = com.atlassian.jira.component.ComponentAccessor.issueManager
    def jIssue = issueManager.getIssueObject(issue.id as Long)
    def localOrgs = getLocalOrganizations(jIssue, remoteOrgs)
    if (cf.getValue(jIssue) == null) cf.customFieldType.createValue(cf, jIssue, localOrgs)
    else {
        cf.customFieldType.updateValue(cf, jIssue, localOrgs)
    }
}
}
if(firstSync){
    Epic.CreateIssue.create(
        replica,
        issue,
        syncRequest,
        nodeHelper,
        issueBeforeScript,
        remoteReplica,
        traces,
        blobMetadataList) {

        setOrganizations(replica.customFields.Organizations)
    }
}
} else {
    setOrganizations(replica.customFields.Organizations)
}

```

[Product](#)

[About Us](#)

[Release History](#)

[API Reference](#)

[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](#)