

# How to Manage Comment Visibility when Receiving Comments in Zendesk

Last Modified on 04/08/2024 5:45 am EDT

This page describes how to keep internal comments internal and restrict synced comments' visibility.

Exalate allows handing comments visibility based on a user group or role restriction.

- **Outgoing sync - send comments**
  - Jira Software
  - Jira Service management
- **Incoming sync - handle received comments**
  - Jira Software
  - Jira Service management
  - Zenmanagement

## Outgoing sync - send comments

### Jira Software

Jira Software allows limiting comments to users with certain project roles/groups. By default, JIRA is configured to allow restricting comment visibility to project roles only. You can restrict comment visibility to a certain user group or based on the user role.

- Send all comments to the destination side

```
replica.comments = issue.comments
```

- Don't send comments which have the user group/role security set

You can use [filterLocal](#) comment helper to filter out comments that you want to synchronize.

```
// ensure that only public comments are sent over to the remote instance  
replica.comments = commentHelper.filterLocal(issue.comments)
```

- Send all comments except the ones created by a user with the group level **dev**

```
replica.comments = issue.comments.findAll { it.roleLevel != "dev" }
```

### Jira Service management

In Jira Service management you can create comments as internal or public.

- Send all comments to the destination side

```
replica.comments = issue.comments
```

- Send only public comments to the destination side

```
replica.comments = issue.comments.findAll { !it.internal }
```

## Incoming sync - handle received comments

### Jira Software

Jira Software allows limiting comments to users with certain project roles/groups. You can add the incoming comment with a certain group/role level and change the existing comment's group/role level.

- Create local comment based on the remote comment visibility

```
issue.comments = commentHelper.mergeComments(issue, replica, {
  comment ->
    if (comment.internal) {
      // if the remote comment is internal make it visible to only users with role "team"
      comment.roleLevel = "team"
    } else {
      // remove all restrictions
      comment.roleLevel = null
      comment.groupLevel = null
    }
  }
  comment
})
```

### Jira Service management

You can handle incoming comments in different ways:

- Add all comments received from the remote side as internal comments

This approach also works for Zenmanagement comments.

```
issue.comments = commentHelper.mergeComments(issue, replica, {it.internal = true; it})
```

- Add comments received from the remote side as internal comments based on a specific group role

```
issue.comments = commentHelper.mergeComments(issue, replica, {
  // make the comment internal, if the comment is visible only to the user role "team"
  comment ->
    comment.internal = comment.roleLevel == "team"
  // collect the comments
  comment
})
```

### Zendesk

- Add all incoming comments as Internal notes to the Zendesk tickets

```
issue.comments = commentHelper.mergeComments(issue, replica, {
  comment ->
    comment.internal = true
  comment
}
)
```

#### Product

[About Us](#)

#### ON THIS PAGE

[Release History](#)

[Glossary](#)

[Outgoing sync - send comments](#)

[API Reference](#)

[Incoming sync - handle received comments](#)

[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](#)