

How to Sync Rich Text and HTML Between Freshdesk and Other Systems

Last Modified on 03/31/2025 10:36 am EDT

When syncing data between **Freshdesk** and other systems like **Jira**, **GitHub**, **Azure DevOps**, or **Jira Cloud**, it's important to manage the **rich text formatting** to ensure that the content maintains its appearance and structure across different platforms. Freshdesk, by default, supports HTML formatting, which offers flexibility in how information is presented. However, there are specific considerations when syncing rich text content, especially when dealing with systems that support **Markdown** vs. **HTML** formatting.

Freshdesk Formatting Capabilities

Freshdesk supports **HTML formatting** for its descriptions and comments, which allows for various rich text elements such as:

- **Bold**
- **Italic**
- **Underlined**
- **Horizontal rule**
- **Ordered lists**
- **Unordered lists**
- **External links**
- **Internal links**

Known Limitations

- **Inline attachments**, like images, are **not supported**.
- Rich text can only be used in **descriptions** and **comments**. Multi-line custom fields **do not support rich text**.

In Exalate, the handling of rich text formatting depends on whether you want to preserve the formatting across systems or whether a **plain text** version is acceptable. Here's a breakdown of

when to use different configurations:

1. Default Configuration (Plain Text)

If you do not need to preserve any rich text formatting and prefer to sync the data as plain text across systems, it's safest to keep the `stripHtml` nodeHelpers for description and comments as they are provided in the default Freshdesk script rules. This will ensure that the rich text formatting (HTML) is stripped out before the data is sent to the destination system, keeping the synchronization simple and compatible across all systems.

Freshdesk outgoing

```
replica.key      = entity.key
replica.summary  = entity.summary
replica.description = nodeHelper.stripHtml(entity.description)
replica.status   = entity.status
replica.priority = entity.priority
replica.reporter = entity.reporter
replica.comments = nodeHelper.stripHtmlFromComments(entity.comments)
replica.attachments = entity.attachments
```

// Exalate API Reference Documentation: <https://docs.exalate.com/docs/exalate-api-reference-documentation>

2. Syncing HTML Formatting Between Systems That Support HTML

If both systems you are syncing between support **HTML formatting**, such as **FreshDesk**, **Azure DevOps (ADO)**, and **GitHub**, you can preserve the rich text formatting during synchronization. To achieve this **remove** the `stripHtml` nodeHelpers from the **outgoing** sync scripts in FreshDesk.

```
replica.key      = entity.key
replica.summary  = entity.summary
replica.description = entity.description
replica.status   = entity.status
replica.priority = entity.priority
replica.reporter = entity.reporter
replica.comments  = entity.comments
replica.attachments = entity.attachments
```

// Exalate API Reference Documentation: <https://docs.exalate.com/docs/exalate-api-reference-documentation>

3. Syncing Between Freshdesk and Markdown-Supporting Systems (like Jira)

When syncing between Freshdesk and systems like **Jira** that support **Markdown** but **not HTML**:

- **Remove** the `stripHtml` node helper from the **outgoing** sync scripts in Freshdesk:

```
replica.key      = entity.key
replica.summary  = entity.summary
replica.description = entity.description
replica.status   = entity.status
replica.priority = entity.priority
replica.reporter = entity.reporter
replica.comments = entity.comments
replica.attachments = entity.attachments
```

// Exalate API Reference Documentation: <https://docs.exalate.com/docs/exalate-api-reference-documentation>

- On the **destination side (e.g. Jira)**, add the node helper `toMarkDownFromHtml` as explained in the Exalate [documentation](#). This will convert the HTML formatting from FreshDesk into the appropriate Markdown syntax on Jira:

```
issue.description = nodeHelper.toMarkDownFromHtml(replica.description)
issue.comments    = nodeHelper.toMarkDownComments(commentHelper.mergeComments(issue, replica))
```

Product

APIs

Release History

Client

APIs

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