How to Sync Custom Fields in Freshservice

This article shows how to synchronize custom fields in Freshservice.

Exalate supports the following types of custom fields from Freshservice:

- Single line

- Multi line

- Multi line dropdown

- Checkbox

- Dropdown

- Date

- Number

- Decimal

- Dependent fields

> **Known limitation:** updating Freshservice date, dropdown, checkbox custom fields does not update the corresponding value on the destination side.

When creating a new custom field in Freshservice, you can define both the Label for Agents and the Label for Customers. However, when writing scripts or using custom fields in Exalate, it is important to use the generated name derived from the **Label for Agents**. The naming convention follows a specific set of rules.

Freshservice Custom Field Naming Rules

When a custom field is created in Freshservice, the name used in Exalate must follow these rules:

1. **Prefix**: All custom field names must begin with `cf_`.

2. **Lowercase Conversion**: The label is automatically converted to lowercase.

3. **Spaces Replaced by Underscores**: Any spaces in the label are replaced with underscores (`_`).

4. **Special Characters Removed**: Special characters (such as `@`, `#`, `$`, `%`, `&`, `*`, etc.) are stripped out.

5. **Restricted Keywords**: Certain system-reserved words may not be allowed as part of the custom field name.

Naming Examples

Here are some examples of how Freshservice field labels translate into generated names:

Label (Agent)	Generated Name
"Label Agent"	cf_label_agent
"Customer ID"	cf_customer_id
"Order #"	cf_order
"User Email@"	cf_user_email
"This is a very#very@long Text name"	cf_this_is_a_veryverylong_text_name

How to Define Custom Fields in Freshservice

To create or edit custom fields in Freshservice, refer to the Freshservice documentation for the full process and guidance. Here's the relevant section from Freshservice: **Customizing Ticket Fields with Field Manager**

There are always two main configuration points: *Outgoing sync,* which is the sending side, and *Incoming sync,* which is the receiving side.

Outgoing Sync

These are examples of how to send the value of the Freshservice custom field to the destination custom field.

```
replica.customFields."Single Line"        = entity.customFields."single_line"
replica.customFields."Multi Line"         = entity.customFields."multi_line"
replica.customFields."Multi Line Dropdown"  = entity.customFields."msf_multi_dropdown"
replica.customFields."Number"             = entity.customFields."number"
replica.customFields."Decimal"            = entity.customFields."decimal"
replica.customFields."Dropdown"            = entity.customFields."dropdown"
replica.customFields."Checkbox"           = entity.customFields."checkbox"
replica.customFields."Date"             = entity.customFields."date"
replica.customFields."Dependent Field L1"     = entity.customFields."dependent_field"
replica.customFields."Dependent Field L2"     = entity.customFields."df_level_2"
replica.customFields."Dependent Field L3"     = entity.customFields."df_level_3"
```

Incoming Sync

These are examples of how to assign the value of the received custom field to the Freshservice custom field.

```
entity.customFields."single_line"?.value = replica.customFields."Single Line"?.value
entity.customFields."multi_line"?.value = replica.customFields."Multi Line"?.value
entity.customFields."msf_multi_dropdown"?.value = replica.customFields."Multi Dropdown"?.value?.value
entity.customFields."number"?.value = replica.customFields."Number"?.value
entity.customFields."decimal"?.value = replica.customFields."Decimal"?.value
entity.customFields."dropdown"?.value = replica.customFields."Dropdown"?.value?.value
entity.customFields."checkbox"?.value = true // My Jira project doesn't have a single checkbox with boolean value
entity.customFields."date"?.value = replica.customFields."Date"?.value
```