Chapter No. 4
"Field Management"
In this package, you will find:
A Biography of the author of the book
A preview chapter from the book, Chapter NO.4 "Field Management"
A synopsis of the book’s content
Information on where to buy this book

About the Author

Patrick Li is the co-founder and senior engineer of AppFusions. AppFusions is the leading Atlassian partner, specializing in integration solutions with many enterprise applications and platforms, including IBM Connections, Jive, Google Apps, Box, SugarCRM, and more.

He has worked in the Atlassian ecosystem for over five years, developing products and solutions for the Atlassian platform, and providing expert consulting services. He is one of the top contributors to the Atlassian community, providing answers and suggestions on the Atlassian user forum.

He has extensive experience in designing and deploying Atlassian solutions from the ground up, and customizing existing deployments for clients across verticals such as healthcare, software engineering, financial services, and government agencies.

I would like to thank my wife, Katherine, who supported and encouraged me along the way, especially during my relocation to San Francisco from Sydney. I would also like to thank all the reviewers for their valuable feedback, and also the publishers/ coordinators, for their help in making this happen.

For More Information:  
JIRA 5.2 Essentials

JIRA 5 was unveiled during the Atlassian Summit in 2011, the annual conference event that brings together users, partners, and experts around the world. It was at this time that we were given the first glimpse of the advancements and new direction that Atlassian is taking JIRA, the world's most popular issue-tracking software. Fast forward to 2012, and Atlassian announced its JIRA Enterprise offering and shortly after, JIRA 5.1, another monumental milestone for this amazing software. With this new release, Atlassian took JIRA to the next level, transforming it from a single software package into a complete platform that supports agile practices through plugins such GreenHopper and Bonfire, adding social aspects through features such as sharing and mentions, and a much improved user experience from the interface down to its core performance.

While JIRA 5 introduces numerous new features and enhancements, some of the more prominent changes for the old-time users are on the newly improved user interface designs. For this reason, we will assume nothing, and introduce JIRA fresh from the perspective of JIRA 5.

Packed with real-life examples and step-by-step instructions, this book will help you become a JIRA expert.

What This Book Covers

This book is organized into eleven chapters. Chapter 1, Getting Started with JIRA, starts with setting up your own JIRA, and the subsequent chapters will introduce the key features and concepts. With each chapter, you will learn important concepts such as business processes, workflows, e-mails, and notifications, and you will have the opportunity to put your newly acquired knowledge into practice by following a live JIRA sample implementation.

Chapter 1, Getting Started with JIRA, serves as a starting point for the book and aims to guide you through setting up a local copy of the JIRA application that will be used throughout the book. For seasoned JIRA experts, this will both refresh your knowledge and also introduce you to the new changes in JIRA 5. By the end of the chapter, you should have a running JIRA application.

Chapter 2, Project Management, covers how to set up projects and project-related administration tasks in JIRA. The concept of schemes will also be introduced, as it is the core concept in JIRA administration.

For More Information:
Chapter 3, *Issue Management*, covers everything related to issue creation and operations that can be performed on an issue (excluding workflow transitions). Furthermore, this chapter will gently touch on the various aspects of issues, as they are the focal point of JIRA. This chapter will also serve as an opportunity to show and allow you to set up dummy data that will be used by the sample project.

Chapter 4, *Field Management*, covers how JIRA collects data through the use of fields and how to expand on this ability through the use of custom fields. The chapter will then continue with the various behaviors that can be configured for fields.

Chapter 5, *Screen Management*, builds on the preceding chapter and explores the concept of screens and how users can create and manage their own screens. This chapter will tie in with all the previous chapters to show the power behind JIRA's screen design capabilities.

Chapter 6, *Workflows and Business Processes*, explores the most powerful feature offered by JIRA, workflows. The concept of issue life cycle will be introduced and various aspects of workflows explained. This chapter will also explore the relationship between workflows and other JIRA aspects previously covered, such as screens. The concept of JIRA extensions will also be briefly touched on in the sample project, using some popular free extensions.

Chapter 7, *E-mails and Notifications*, focuses on how to get automatic e-mail notifications from JIRA and explores the different settings that can be applied. This is a very important and powerful feature of JIRA and also a critical part of the example project for this book. This chapter will also tie in with the workflow chapter and explain in detail how JIRA manages its notification mechanism.

Chapter 8, *Securing JIRA*, focuses on the different security control features offered by JIRA. As this topic affects all aspects of JIRA, all previous topics will be touched on, explaining how security can be applied to each. We will also cover LDAP integration where you can hook up your JIRA with an existing LDAP system for user management.

Chapter 9, *Searching, Reporting, and Analysis*, will focus on how data captured in JIRA can be retrieved to provide various types of reporting features. We will also introduce the new search features introduced in JIRA 5.

Chapter 10, *General Administration*, covers other administration features offered by JIRA, including using add-ons and the Atlassian Marketplace. These features often do not form the backbone of a JIRA installation but can be very useful when used properly.

Chapter 11, *Advanced Features*, covers advanced features that can help to change your JIRA into more than just a traditional issue-tracking system. We will look at how you can run agile projects with JIRA through the use of GreenHopper, and turn JIRA into an effective feedback collection system.

---

For More Information:

Projects are collections of issues, and issues are collections of fields. As we have seen in the earlier chapters, fields are what capture and display data to users. There are many different types of fields in JIRA, ranging from simple text fields that let you input alphanumeric texts, to more complicated fields with pickers to assist you in choosing dates and users.

An information system is only as useful as the data that goes into it. By understanding how to effectively use fields, you can turn JIRA into a powerful information system for data collection, processing, and reporting.

In this chapter, we will expand on our Help Desk project with these customized fields and configurations, by exploring fields in detail and learn how they relate to other aspects of JIRA. By the end of this chapter, you will have learned the following:

- Understand built-in and custom fields, and what they are
- Extend JIRA's ability to collect data through custom fields
- Adding behaviors to fields with field configurations
- Field configuration schemes and how to apply them to projects

For More Information:  
Built-in fields
JIRA comes with a number of built-in fields. You have already seen a few of them in the previous chapters. Fields, such as summary, priority, and assignee, are all built-in. They make up the backbone of an issue, and you cannot remove them from the system. For this reason, they are referred to as system fields. The following table lists the most important built-in fields in JIRA:

<table>
<thead>
<tr>
<th>System field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignee</td>
<td>Specifies the user who is currently assigned to work on the issue.</td>
</tr>
<tr>
<td>Summary</td>
<td>Specifies a one-line summary of the issue.</td>
</tr>
<tr>
<td>Description</td>
<td>Provides a detailed description of the issue.</td>
</tr>
<tr>
<td>Reporter</td>
<td>Specifies the user who has reported this issue (although most of the time it is also the person who has created the issue, but not always).</td>
</tr>
<tr>
<td>Component/s</td>
<td>Specifies the project components the issue belongs to.</td>
</tr>
<tr>
<td>Effects Version/s</td>
<td>Specifies the versions that the issue effects are found in.</td>
</tr>
<tr>
<td>Fix Version/s</td>
<td>Specifies the versions that the issue will be fixed in.</td>
</tr>
<tr>
<td>Due Date</td>
<td>Specifies the date when this issue is due.</td>
</tr>
<tr>
<td>Issue Type</td>
<td>Specifies the type of the issue (for example, Bug and New Feature).</td>
</tr>
<tr>
<td>Priority</td>
<td>Specifies how important the issue is compared to other issues</td>
</tr>
<tr>
<td>Resolution</td>
<td>Specifies the current resolution value of the issue (for example, Unresolved or Fixed).</td>
</tr>
<tr>
<td>Time Tracking</td>
<td>Lets users specify estimates for how long the issue will take to complete.</td>
</tr>
</tbody>
</table>

Custom fields
While JIRA's built-in fields are quite comprehensive for basic general uses, most organizations will soon find that they have special requirements that cannot be addressed simply with the default fields available. To help you tailor JIRA to your organization's needs, JIRA lets you create and add your own fields to the system, called custom fields.

Custom field types
Every custom field belongs to a custom field type, which dictates its behavior, appearance, and functionality. So, when you add a custom field to JIRA, you really add another instance of a custom field type.
JIRA comes with over 20 custom field types that you can use straight out of the box. Many of the custom field types are identical to the built-in fields, but provide you with more control and flexibility that are not available with their built-in counterparts. The following tables breaks down and lists all the JIRA standard custom field types and their characteristics.

**Simple fields**
These fields are the most basic field types in JIRA. They are usually simple and straightforward to use, such as text field, which allows users to input any text:

<table>
<thead>
<tr>
<th>Custom field type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free text field (unlimited text)</td>
<td>These are multiple line text-areas enabling entry for large text contents.</td>
</tr>
<tr>
<td>Number field</td>
<td>These are input fields that store and validate numeric values.</td>
</tr>
<tr>
<td>Radio buttons</td>
<td>These are radio buttons that ensure only one value can be selected.</td>
</tr>
<tr>
<td>Select list</td>
<td>These are single select lists with a configurable list of options.</td>
</tr>
<tr>
<td>Text field</td>
<td>These are basic single link input fields to allow simple text inputs of less than 255 characters.</td>
</tr>
<tr>
<td>URL field</td>
<td>These are input fields that validate a valid URL.</td>
</tr>
</tbody>
</table>

**JIRA specialized fields**
These fields provide specialized functions. For example, the Date Picker field provides you with a calendar to let you pick a date from, and User Picker has an auto-complete feature to help you find the user you want to select:

<table>
<thead>
<tr>
<th>Custom field type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascading select</td>
<td>These are multiple select lists where the options for the second select list are dynamically updated based on the value of the first.</td>
</tr>
<tr>
<td>Date picker</td>
<td>These are input fields that allow input with a date picker and enforcing valid dates.</td>
</tr>
<tr>
<td>Date time</td>
<td>These are input fields that allow inputs with a date and time picker and enforcing valid date timestamps.</td>
</tr>
<tr>
<td>Group picker</td>
<td>These choose a user group using a pop-up picker window.</td>
</tr>
<tr>
<td>Labels</td>
<td>These are input fields that allow labels to be added to an issue.</td>
</tr>
</tbody>
</table>

For More Information:
Field Management

<table>
<thead>
<tr>
<th>Custom field type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project picker</td>
<td>These select lists displaying the projects viewable by the user in the system.</td>
</tr>
<tr>
<td>Read-only text field</td>
<td>These are read-only text fields that do not allow users to set their data. It's only possible to programmatically set the data.</td>
</tr>
<tr>
<td>Single version picker</td>
<td>These choose a single version from the available versions in the project.</td>
</tr>
<tr>
<td>User picker</td>
<td>These choose a user from the JIRA user base via either a pop-up user picker window or through auto completion.</td>
</tr>
<tr>
<td>Version picker</td>
<td>These choose one or more versions from the available versions in the current project.</td>
</tr>
</tbody>
</table>

Multi-value fields

These fields are like their single value versions and allow you to select multiple values rather than only one. For example, the Multi Select field lets you select one or more values, while its singular equivalent, the Select List field, only lets you choose one. The only exception is the Multi Checkboxes field, which does not come with a singular variety:

<table>
<thead>
<tr>
<th>Custom field type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi checkboxes</td>
<td>These are checkboxes that allow multiple values to be selected.</td>
</tr>
<tr>
<td>Multi group picker</td>
<td>They choose one or more user groups using a pop-up picker window.</td>
</tr>
<tr>
<td>Multi select</td>
<td>They select lists allowing multiple values to be selected.</td>
</tr>
<tr>
<td>Multi user picker</td>
<td>They choose one or more users from the user base via a pop-up picker window.</td>
</tr>
</tbody>
</table>

As you can see, JIRA provides you with a comprehensive list of custom field types. In addition, there are many custom field types developed by third-party vendors (called plugins or add-ons) that you can add to your JIRA to enhance its functionality. These custom fields provide many specialized functionalities, such as automatically calculating values, retrieving data from databases directly, or connecting to an external system. Once you have installed the plugin, the process of adding custom fields from other vendors is mostly the same as adding custom fields shipped with JIRA.

For More Information:
We will look at plugins in Chapter 10, General Administration, and discuss where to find plugins, and how to install and manage them. The following list shows some examples of plugins that provide additional useful custom fields:

- **JIRA Enhancer Plugin**: This includes a number of custom fields that will automatically display dates when key events occur for an issue. For example, when the issue was last closed.

- **JIRA Toolkit Plugin**: This provides a number of useful custom fields, such as automatically showing you all the users that participate in a given issue, and the date when the issue was last commented on.

- **nFeed**: This provides a suite of custom fields that let you connect to databases, remote files, and web services to retrieve data and display them in JIRA.

- **CFR Part 11 E-Signature**: This lets users electronically sign issues in JIRA as they work on them; for example, approving an issue to be closed.

- **SuggestiMate for JIRA**: This provides a specialized custom field that shows similar and potentially duplicated issues when you are creating new fields or browsing existing ones.

### Searchers

For any information system, capturing data is only half of the equation. Users will need to be able to retrieve the data at a later stage, usually through searching, and JIRA is no different. While fields in JIRA are responsible for capturing and displaying data, it is their corresponding searchers that provide the search functionality.

In JIRA, all the built-in fields have searchers associated by default, so you will be able to search issues by their summary or assignee, without any further configuration. For custom fields, however, you will need to specify a searcher for each custom field you add. If you do not specify a searcher, you will not be able to search data based on that field.

For all custom field types that come with JIRA, one or more searchers are available for you to choose from. You can select a searcher when you create a new custom field and later change the searcher, as we will see when we cover how to manage custom fields.

For More Information:
Field Management

Custom field context

Built-in fields, such as priority and resolution, are static across JIRA. What this means is that these fields will have the same set of selections for all projects. Custom fields, on the other hand, are a lot more flexible.

Custom field types, such as select list and radio buttons, can have different sets of options for different projects, or different issue types within the same project. This is achieved via what is called **custom field context**.

A custom field context is made up of a combination of applicable projects and applicable issue types. When you are working with an issue, JIRA will check the project and issue type of the current issue to determine if there is a specific context that matches the combination. If one is found, JIRA will load the custom field with any specific settings such as selection options. However, if no context is found, the custom field will not be loaded.

In JIRA, if no context can be found that matches the project and issue type combination, the custom field does not exist for the said issue.

We will look at how to set custom field contexts in a later section. What you need to remember now is when adding a custom field, you need to make sure that it has the correct context setting.

Managing custom fields

Just like built-in fields, custom fields are used globally across JIRA, so you will need to have the JIRA Administrator global permission to carry out management operations, such as creation and configuration.

JIRA maintains all the custom fields in a centralized location for easy management. Perform the following steps to access the custom field management page:

1. Log in to JIRA as a member of the **jira-administrator** group.
2. Click on **Administration** in the top menu bar.
3. Select **Custom Fields** from underneath the **Fields...** section:

For More Information:

On the **Custom Fields** page, all existing custom fields will be listed. From here, you can see the name of each custom field, their type, the context they belong to, and the screens they are on.

### Adding a custom field

Creating a new custom field is a multi-step process, and JIRA provides a wizard to help you through the process. There are two required steps and an optional step when adding a new custom field. You need to first determine the type of custom field, and then its searcher and context. The last optional step is to decide what screens to add the field onto. We will be walking through the following process:

1. **Browse to the Custom Fields page.**
2. **Click on the Add Custom Field button.** This will bring you to step 1 of the process where you can select the custom field type.
3. Select the custom field type you wish to add and click on **Next**. This will bring you to step 2 of the process, where you can specify other aspects of the new custom field:

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug Import Id</td>
<td>A read-only custom field that points back to the previously imported bug id.</td>
</tr>
<tr>
<td>Date Picker</td>
<td>A custom field that stores dates and uses a date picker to view them</td>
</tr>
<tr>
<td>Free Text Field (unlimited text)</td>
<td>A multline text area custom field to allow input of longer text strings.</td>
</tr>
<tr>
<td>Hidden Job Switch</td>
<td>Hidden switch programmatically set whether or not to create a Perforce job.</td>
</tr>
<tr>
<td>Job Checkbox</td>
<td>Checkbox to choose whether or not to create a Perforce job.</td>
</tr>
<tr>
<td>Multi Checkboxes</td>
<td>Choose multiple values using checkboxes.</td>
</tr>
<tr>
<td>Multi Select</td>
<td>Choose multiple values in a select list.</td>
</tr>
<tr>
<td>Number Field</td>
<td>A custom field that stores and validates numeric (floating point) input.</td>
</tr>
<tr>
<td>Radio Buttons</td>
<td>A list of radio buttons.</td>
</tr>
<tr>
<td>Select List</td>
<td>A single select list with a configurable list of options.</td>
</tr>
<tr>
<td>Text Field (+255 characters)</td>
<td>A text line text box custom field to allow single text input.</td>
</tr>
<tr>
<td>User Picker</td>
<td>Choose a user from the user base via a popup picker window.</td>
</tr>
</tbody>
</table>

4. Enter values for the **Field Name** and **Field Description** fields. You can have multiple custom fields with the same name. However, doing so will make it harder for maintenance.

5. Select a search template, if available. All custom field types that are shipped with JIRA will have one or more search templates. If you do not wish the field to be searchable, then select **None**.

6. Select which issue types the custom field will be available for. If you select the **All issue type** option, the custom field will be available to all the issue types (assuming if it is also available for the project).

---

**For More Information:**

7. Select the projects the custom field will be available for. **Global context** means all the projects are in JIRA. If you select specific projects, then the custom field will only be available for issues within those projects (if it is available for the issue type):

8. Click on **Finish**. This will bring you to the last step of the process, where you can specify which screen you would like to add the field onto. This step is optional, as the custom field has already been added in JIRA. You do not have to add the field onto a screen. We will be discussing fields and screens in Chapter 5, **Screen Management**.

---

For More Information:

9. Select the screens and click on Update:

Once a custom field has been created, you will be able to manage its configurations and settings.

**Editing/deleting a custom field**

Once a custom field has been created, you can edit its details at any time. You might have already noticed that there is a **Configure** option and an **Edit** option for each custom field. It can be confusing in the beginning to differentiate between the two. Configure specifies options related to custom field context, which we will discuss in the following sections. Edit specifies options that are global across JIRA for the custom field; these include its name, description, and search templates:

1. Browse to the **Custom Fields** page.
2. Select the **Edit** option by clicking on the tools icon for the custom field you wish to edit from the list of custom fields.
3. Change the custom field details.
4. Click on the **Update** button to apply the changes.
When making changes to the search templates for your custom fields, it is important to note that while the change will take effect immediately, you need to perform a full system reindex in order for JIRA to return correct search results. This is because for each search template, the underlying search data structure might be different, and JIRA will need to update its search index for the newly applied search template.

For example, if you have a custom field that did not have a searcher and you have just applied a searcher to it (while the searcher will now be available), no results will be returned until you reindex JIRA. When you make changes to the search template, JIRA will alert you with a message that a reindex will be required.

Prior to JIRA 5.2, reindex will make JIRA inaccessible until the process completes, and it can be time consuming for large instances. So, make sure you schedule a maintenance window to minimize the impact to your users. Starting with 5.2, JIRA lets you reindex in the background.

We will be discussing searching and indexing in more detail in Chapter 9, Searching, Reporting, and Analysis.

You can also delete the existing custom fields, as follows:

1. Browse to the Custom Fields page.
2. Select the Delete option by clicking on the tools icon for the custom field you wish to delete.
3. Click on the Delete button to delete the custom field.

Once deleted, you cannot get the custom field back and you will not be able to retrieve and search the data held by those fields. If you try to create another custom field of the same type and name, it will not inherit the data from the previous custom field, as JIRA assigns unique identifiers to each of them. It is highly recommended to back up your production JIRA before you delete the field; we will discuss backup strategies in Chapter 10, General Administration.
Configuring a custom field

Now that we have seen how to create and manage custom fields, we can start looking at the more advanced configuration options. Different custom field types will have different configuration options available to them. For example, while all custom fields will have the option to specify one or more contexts, selecting list-based custom fields will also allow you to specify a list of options. We will look at each of the configuration options in the following sections.

To configure a custom field, you need to access the Configure Custom Field page, as follows:

1. Browse to the Custom Fields page.
2. Select the Configure option by clicking on the tools icon for the custom field you wish to configure from the list of custom fields. This will bring you to the Configure Custom Field page.

The following screenshot shows that the Department custom field has two available contexts, the default context (Default Configuration Scheme for Department), and Support Context, which is applied only to the Support Desk project:

For More Information:
Adding custom field contexts

From time to time, you may need your custom fields to have different behaviors depending on what project the issue is in. For example, if we have a select list custom field called **Department**, we may want it to have a different set of options based on which project the issue is being created, or even a different default value.

To achieve this level of customization, JIRA allows you to create multiple custom field contexts for a custom field. As we have seen already, a custom field context is a combination of issue types and projects. So, in our example, we can create a context for issue type **Bug** and project **Support**, and set the default department to **Engineering**.

---

JIRA allows you to configure custom fields based on issue types and projects through contexts. Each project can have only one configuration context per custom field.

---

Creating a new custom field context is simple. All you need to do is decide the issue type and project combination that will define the context:

1. Browse to the **Configure Custom Field** page for the custom field you wish to create a new context for.
2. Click on the **Add new context** link. This will take you to the **Add configuration scheme context** page.
3. Give a name to the new custom field context in the **Configuration scheme** label field.
4. Select the applicable issue types.
5. Select the applicable projects.
6. Click on the **Add** button to create the new custom field context.

Each project can only belong to one custom field context per custom field (global context is not counted for this). Once you have selected a project for a context, it will not be available the next time you create a new context. For example, if you have created a new context for **Project A**, it will not be listed as an option when you create another context for the same custom field. This is to prevent you from accidentally creating two contexts for the same project.
After a new custom field context has been created, it will not "inherit" any configuration values such as Default Value and Select Options from other contexts, such as the default context. You will need to repopulate and maintain the configuration options for each newly created context:

**Configuring select options**

For custom fields of types select list, checkboxes, radio buttons, and their multi versions, you need to configure their select options before they can become useful to the users. The select options are configured and set on a per custom field context basis. This provides the custom field with the flexibility to have a different set of options for a different context.

For More Information:
To configure the select options, you need to first select the custom field and then the context that the options will be applied to, as follows:

1. Browse to the Custom Fields page.
2. Click on the Configure option for the custom field you wish to configure select options for.
3. Click on the Edit Options link for the custom field context to apply the options for.
4. Fill in the option values in the Add New Custom Field Option section, and click on the Add button to add the value. The options will be added in the order they are entered into the system. You can manually move the option values up and down or click on Sort options alphabetically to let JIRA perform the sorting for you.
5. Click on the Done button once you have finished configuring the select options:

<table>
<thead>
<tr>
<th>Position</th>
<th>Option</th>
<th>Order</th>
<th>Move To Position</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Customer Service</td>
<td>$✓  $✓</td>
<td></td>
<td>Edit</td>
</tr>
<tr>
<td>2</td>
<td>Engineering</td>
<td>$✓  $✓</td>
<td></td>
<td>Edit</td>
</tr>
<tr>
<td>3</td>
<td>P&amp;T</td>
<td>$✓  $✓</td>
<td></td>
<td>Edit</td>
</tr>
<tr>
<td>4</td>
<td>Sales</td>
<td>$✓  $✓</td>
<td></td>
<td>Edit</td>
</tr>
<tr>
<td>5</td>
<td>Support</td>
<td>$✓  $✓</td>
<td></td>
<td>Edit</td>
</tr>
</tbody>
</table>

Setting default values
For most custom fields, you can set a default value so that your users will not need to fill them in unless they have special needs. For text-based custom fields, the default values will be displayed as text by default, when the users create or edit an issue. For selection-based custom fields, the default values will be options pre-selected for the users.
Field Management

Just like setting selection options, default options are also set on a per-custom field context basis:

1. Browse to the Custom Fields page.
2. Click on the Configure option for the custom field you wish to configure select options for.
3. Click on the Edit Default Value link for the custom field context to apply the default values for.
4. Set the default value for the custom field.
5. Click on the Set Default button to set the default value.

The way to set the default value will be different for different custom field types. For text-based custom fields, you will be able to type any text string. For select-based custom fields, you will be able to select from the options you have added (if you have not set any options for the custom field, the only option available as a default value will be None). For picker-based custom fields, such as user picker, you will be able to select a user directly from the user base:

Field configuration
As you have already seen, fields are used to capture and display data in JIRA. Fields can also have behaviors, which are defined by field configuration. For each field in JIRA, you can configure its behaviors listed as follows:

- Field description: This is the description text that appears under the field when an issue is edited.
- Visibility: This determines if a field should be visible or hidden.
- Required: This specifies if a field will be optional or required to have a value when an issue is being created/updated. When applied to a select, checkbox, and radio button custom fields, this will remove the None option from the list.
- Rendering: This specifies how the content is to be rendered for text-based fields (for example, wiki renderer or simple text renderer for text fields).

For More Information:
A field configuration provides you with control over each individual field in your JIRA, including both built-in and custom fields. Since it is usually a good practice to re-use the same set of fields instead of creating new ones for every project need, JIRA allows us to create multiple field configurations, with which we can specify different behaviors on the same set of fields and apply them to different projects.

We will be looking at how to manage and apply multiple field configurations in the later sections in this chapter. But first, let’s take a close look at how to create new field configurations and what we can do with these configurations.

You can access the field configuration management page through the JIRA Administration console:

1. Log in to JIRA as a JIRA Administrator.
2. Click on Administration at the top menu bar.
3. Select Field Configurations under the Fields... section. This will bring you to the View Field Configurations page:

### Adding a field configuration

Creating new field configurations is simple. All you need to do is specify the name and a short description for the new configuration:

1. Browse to the View Field Configurations page.
2. Specify the name for the new field configuration in the Add Field Configuration section.
3. Provide a short description for the field configuration.
4. Click on the Add button to create a field configuration.
Field Management

As we will see later in the Field Configuration Scheme section, field configurations are linked to issue types, so it is recommended to name them based on the issue type they will be applied to and with a version number at the end; for example, Bugs Field Configuration 1.0. This way, when you need to make changes to the field configuration, you can increment the version number, leaving a history of changes that you can revert back to.

After a field configuration is created, it is put into what we call the inactive state. This means that the configuration is not being used anywhere in JIRA and you are free to edit and delete it. In order to activate the field configuration, we need to associate it with a field configuration scheme. We will look at how to do this in later sections.

Editing/deleting a field configuration

You can update existing field configuration details and delete them all together. The details you can edit are the configuration's name and description:

1. Browse to the View Field Configurations page.
2. Click on the Edit link for the field configuration you wish to edit. This will take you to the Edit Field Configuration page.
3. Update the Name and Description fields with new values.
4. Click on the Update button to apply the changes.

You will be able to edit field configuration details at anytime. However, for deletion, you can only delete the configuration when it is inactive. Once you have associated the configuration with a scheme, which will put the configuration into the active state, you cannot delete it until it is back in the inactive state. For you to put the field configuration back into the inactive state, you need to unassociate it from the field configuration scheme:

1. Browse to the View Field Configurations page.
2. Click on the Delete link for the field configuration you wish to delete. This will take you to the Delete Field Configuration page for confirmation.
3. Click on the Delete button to delete the field configuration.
Copying a field configuration
A field configuration contains configuration details for all fields in JIRA. For a moderately complicated instance, you are likely to have over 20 fields. It will be very unproductive if you have to reconfigure every single field again, whenever you need to create a new set of field configurations, usually with only minor differences for a few fields.

To simplify your task, JIRA allows you to copy an existing field configuration and use that as a base for you to make only the necessary changes. This greatly reduces the amount of effort required, as you will not have to reconfigure all the fields that are common across all the use cases:

1. Browse to the View Field Configurations page.
2. Click on the Copy link for the field configuration you wish to copy. This will take you to the Copy Field Configuration page.
3. Specify a new name for the field configuration.
4. Specify a description.
5. Click on the Copy button to copy the field configuration.

Managing field configurations
Now that we have seen how to create, edit, delete, and copy field configurations, it is time for us to take a closer look at the different configuration options. Just a quick recap—each field configuration includes all fields available in JIRA and their behaviors are defined specifically to each field configuration. We will then set a context for the field configurations through the use of the field configuration scheme, which will determine when a field configuration will become active for a given issue.

Perform the following steps to access the field configuration options:

1. Browse to the View Field Configurations page.
2. Click on the Configure link for the field configuration you wish to configure. This will take you to the View Field Configuration page (note singular).
On this page, all the fields and their current configuration options that are currently set for the selected field configuration are listed:

As you can see, there are several options you can configure for each field, and depending on the field type, the options may vary. While we will be looking at each of the options, it is important to note that some of the options will override each other. This is JIRA trying to protect you from accidentally creating a configuration combination that will break your JIRA. For example, if a field is set to both hidden and required, your users will not be able to create or edit issues, so JIRA will not allow you to set a field to required if you have already set it to hidden.

**Field description**

While having a meaningful name for your fields will help your users to understand what the fields are for, providing a short description will provide more context and meaning. Field descriptions are displayed under the fields when you create or edit an issue. To add a description for a field, do the following:

1. Browse to the View Field Configuration page for the field configuration you wish to use.
2. Click on the Edit link for the field you wish to set a description for.
3. Add the description for the field, and click on Update.
Field requirement

You can set certain fields as required or compulsory for issues. This is a very useful feature as it ensures that critical information can be captured when users create issues. For example, for our support system, it makes sense to have our users fill in the system that is misbehaving and make that field compulsory to help our support engineers.

You have already seen required fields in action. System fields, such as Summary and Issue Type, are compulsory in JIRA (and you cannot change that). When you do not specify a value for a required field, JIRA will highlight the field in red with an error message telling you that the field is required.

When you add a new field into JIRA, such as custom fields, they are optional by default; meaning users do not need to specify a value. You can then change the setting to make those fields required:

1. Browse to the View Field Configuration page for the field configuration you wish to use.
2. Click on the Required/Optional link for the field you wish to set its mandatory requirement.

You will notice that once a field is set to required, there will be a small text label Required in red next to the field name, and when you create or edit an issue, the field will have a red * character next to its name. This is JIRA's way to indicate that a field is required.

Field visibility

Most fields in JIRA can be hidden from users. When a field is set to hidden, users will not see the fields on any screens including issue create, update, and view. Perform the following steps to show or hide a field:

1. Browse to the View Field Configuration page for the field configuration you wish to use.
2. Click the Show/Hide link for the field you wish to show or hide respectively.

You will notice that when a field is set to hidden, its name gets grayed out. Once a field has been set to hidden, it will not appear on screen and you will not be able to search on it.

For More Information:
Field Management

Not all fields can be hidden. Built-in fields, such as **Summary** and **Issue Type**, cannot be hidden. When you set a field to hidden, you will notice that you can no longer set the same field as required. As stated earlier, setting a field to be required will make JIRA enforce a value to be entered into the field when you create or edit an issue. If the field is hidden, there will be no way for you to set a value and you will be stuck. This is why JIRA will automatically disable the required option, if you have already hidden a field. On the other hand, if you have made a field required, when you hide the same field, you will notice that the field is no longer required. So the rule of thumb is that field visibility will override field requirement.

A field cannot be both hidden and required.

Field rendering

Renderers control how a field will look when it is being viewed or edited. Some built-in and custom fields have more than one renderer, and for these fields, you can choose which one to use. For example, for text-based fields, such as **Description**, you can choose to use the default simple text renderer or the more sophisticated wiki style renderer that will allow you to use wiki markup to add more styling.

JIRA ships with four different renderers:

- **Default text renderer**: This is the default renderer for text-based fields. Contents are rendered as plain text. If the text resolves to a JIRA issue key, the renderer will automatically turn that into an HTML link.

- **Wiki style renderer**: This is an enhanced renderer for text-based fields. It allows you to use wiki markup to decorate your text content.

- **Select list renderer**: This is the default renderer for selection-based fields. It is rendered as standard HTML select lists.

- **Autocomplete renderer**: This is an enhanced renderer for selection-based fields and provides an autocomplete feature to assist users as they start typing into the fields.

For More Information:
The following table lists all the fields that can have special renders configured and their available options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Available renderers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Wiki style renderer and default text renderer</td>
</tr>
<tr>
<td>Comment</td>
<td>Wiki style renderer and default text renderer</td>
</tr>
<tr>
<td>Environment</td>
<td>Wiki style renderer and default text renderer</td>
</tr>
<tr>
<td>Component</td>
<td>Autocomplete renderer and select list renderer</td>
</tr>
<tr>
<td>Affects version</td>
<td>Autocomplete renderer and select list renderer</td>
</tr>
<tr>
<td>Fix versions</td>
<td>Autocomplete renderer and select list renderer</td>
</tr>
<tr>
<td>Custom field of type &quot;Free Text Field (unlimited text)&quot;</td>
<td>Wiki style renderer and default text renderer</td>
</tr>
<tr>
<td>Custom field of type &quot;Text Field&quot;</td>
<td>Wiki style renderer and default text renderer</td>
</tr>
<tr>
<td>Custom field of type &quot;Multi Select&quot;</td>
<td>Autocomplete renderer and select list renderer</td>
</tr>
<tr>
<td>Custom field of type &quot;Version Picker&quot;</td>
<td>Autocomplete renderer and select list renderer</td>
</tr>
</tbody>
</table>

Perform the following steps to set the renderer for a field:

1. Browse to the View Field Configuration page for the field configuration you wish to use.
2. Click on the Renderer link for the field you wish to set a renderer for (if it is available). You will be taken to the Edit Field Renderer page.
3. Select the renderer from the available drop-down list.
4. Click on the Update button to set the renderer:

For More Information:
Field Management

There are other custom renderers developed by third-party vendors. Just like custom fields, these are packaged as plugins that you can install in JIRA. Once installed, these custom renderers will be available for selection for the appropriate field types.

A good example is the JEditor plugin, which provides a rich text editor for all text-based fields such as Description.

Screens

In order for a field to appear, it needs to be placed onto a screen. You have already seen this when creating new custom fields. One of the steps in the creation process is to select what screens to add the custom field to. Screens will be discussed further in Chapter 5, Screen Management, so we will not spend too much time understanding it right now. What you need to know for now is, after a field has been added to a screen, you can add it to additional screens or take it off completely. If you are working with just one field, you can configure it here from the field configurations. If you have multiple fields to update, a better approach will be to work directly with screens, as we will see in Chapter 5, Screen Management.

Field configuration scheme

With multiple field configurations, JIRA determines when to apply each of the configurations through the field configuration scheme. A field configuration scheme maps field configurations to issue types. This scheme can then be associated with one or more projects.

This allows you to group multiple field configurations mapped to issue types, and apply them to a project in one go. The project will then be able to determine which field configuration to apply, based on the type of the issue. For example, for a given project, you can have different field configurations for bugs and tasks.

This grouping of configurations into schemes also provides you with the option to reuse existing configurations without duplicating work, as each scheme can be reused and associated to multiple projects.

Managing field configuration schemes

You can manage all your field configuration schemes from the View Field Configuration Schemes page. From there, you will be able to add, configure, edit, delete, and copy schemes:

1. Log in to JIRA as a JIRA administrator.
2. Click on Administration at the top menu bar.

For More Information:
3. Select Field Configuration Schemes at the left-hand side panel. This will bring you to the View Field Configuration Schemes page:

![View Field Configuration Schemes](image)

### Adding a field configuration scheme

The first step to group your field configurations is to create a new field configuration scheme. By default, JIRA does not come with any field configuration schemes. All the projects will use the system default field configuration. The new field configuration scheme will hold all the mappings between our field configurations and issue types.

To create a new field configuration scheme, all you need to do is specify the name and an optional description for the scheme:

1. Browse to the View Field Configuration Schemes page.
2. Fill in a name for the new field configuration scheme in the Add Field Configuration Scheme section.
3. Optionally, provide a short description for the scheme.
4. Click on the Add button to create the scheme.

Since field configuration schemes are applied to projects, it is a good practice to name your them according to the projects. For example, the scheme for the Sales project can be named Sales Field Configuration Scheme. You can add a version number after the name to help you maintain changes.

Once the new field configuration scheme is created, it will be displayed in the table that lists all the existing schemes. At this time, the scheme is in the inactive state, as it does not contain any configuration mappings and is not yet active in JIRA.

---

**For More Information:**

Editing/deleting a field configuration scheme
You can update existing field configuration scheme details and delete them altogether. The details you can edit are the scheme's name and description. You can also update its field configurations mapping, which will be covered in later sections:

1. Browse to the View Field Configuration Schemes page.
2. Click on the Edit link for the field configuration scheme you wish to edit. This will take you to the Edit Field Configuration Scheme page.
3. Update the Name and Description fields with new values.
4. Click on the Update button to apply the changes.

Just like field configurations, you can only delete a field configuration scheme if it is in the inactive state. Once you have associated the scheme with a project, which will put the scheme into the active state, you cannot delete it until it is back in the inactive state. To inactivate a field configuration scheme, you will have to unassociate the scheme from all the projects you have applied it to:

1. Browse to the View Field Configuration Schemes page.
2. Click on the Delete link for the field configuration scheme you wish to delete. This will take you to the Delete Field Configuration Scheme page for confirmation.
3. Click on the Delete button to delete the scheme.

Copying a field configuration scheme
There will be times when you need a new field configuration scheme and the requirements are very similar to a scheme that you already have. So, instead of creating a new scheme from scratch, you can choose to copy the existing scheme as a base, and simply make some quick modifications. JIRA allows you to achieve this by letting you copy the existing schemes:

1. Browse to the View Field Configuration Schemes page.
2. Click on the Copy link for the field configuration scheme you wish to copy. This will take you to the Copy Field Layout Configuration page.
3. Specify the name and description of the new scheme.
4. Click on the Copy button to create a copy.

Once the new-copied scheme is created, you will be able to modify its field configuration and issue type mappings as per your requirements, which we will look at in the next section.

For More Information:
Configuring a field configuration scheme

Once you have a new field configuration scheme setup, you will be able to add mapping between field configurations and issue types. For each field configuration scheme, one issue type can be mapped to only one field configuration, while each field configuration can be mapped to multiple issue types. The following screenshot shows how Development Field Configuration is being applied to both the Technical Task and Improvement issue types, and Bug Field Configuration is applied to the Bug issue type.

One issue type can only be mapped to one field configuration.

When a field configuration scheme is first created, JIRA creates a default mapping, which maps all unmapped issue types to the default field configuration. You cannot delete this default mapping as it acts as a "catch all" condition for mappings that you do not specify in your scheme. What you need to do is to add more specific mappings that will take precedence over this default mapping:

1. Browse to the View Field Configuration Schemes page.
2. Click on the Configure link for the field configuration scheme you wish to configure. This will take you to the Configure Field Configuration Scheme dialog.
3. Select the issue type and field configuration from the Add Issue Type To Field Configuration Association section to establish the mapping.

4. Click on the Add button to add the mapping.

You will notice that once you have added a mapping, the mapped issue type will disappear from the list of selectable issue types. This is JIRA preventing you from double mapping an issue type by accident. Once you have mapped all the available issue types, you will not be able to add any more mappings.

**Associating a field configuration scheme with a project**

After you have created a new field configuration scheme and established the mappings, the configurations will not take effect immediately. The scheme is still in the inactive state. In order to activate the scheme, you need to associate the scheme with a project for the configurations to take effect.

It is important to note that once you have associated the field configuration scheme with a project, you cannot delete it until you have removed all the associations so that the scheme becomes inactive again.

To activate a field configuration scheme, you need to establish the association on a per-project level. This means you need to go to each individual project and set the field configuration scheme option for them:

1. Log in to JIRA as a JIRA Administrator.
2. Click on Administration at the top menu bar.
3. Select the project you wish to associate the field configuration scheme to. This will bring up the Project Administration page.
4. Click on the Fields tab at the left-hand side panel.
5. Select Use a different scheme from the Actions menu. This will bring up the Field Layout Configuration Association page.

For More Information:

6. Select the new field configuration scheme and click on the **Associate** button:

![Field Configuration Scheme Diagram]

You can repeat steps 3 to 6 to associate the field configuration scheme with more projects.

**The Help Desk project**

Now that you have seen how to manage fields in JIRA, it is time to expand on your Help Desk project to include some customized fields and configurations to help your support staff.

What we will do this time is add a few new custom fields to help capture some additional useful data from the business users when they log an incident. We will also create a customized field configuration specially designed for our support team. Lastly, we will tie everything together by associating our fields, configurations, and projects through the field configuration schemes.
Setting up a custom field

Since you are implementing a support system, one common feature is to be able to escalate the incident, and for every escalation, a group of users will be notified automatically. The automatic escalation and notification aspects of this feature will be covered and implemented in later chapters, but what we do need right now is a way to capture the information, such as the following:

- Does the issue require escalation?
- What is the current escalation level?
- Who should be notified when the issue is escalated?

So, to address these requirements, we will be adding three custom fields, one per data requirement.

The first custom field we are going to add is Is Escalation Required. We want to have this option, so that not all tickets raised will require escalation. Some tickets may not be urgent or they are simply for investigation purposes. We will also make this field required, so the users will need to indicate if they require an escalation. To help our users, we will provide a default value of Yes, so tickets by default will require escalation. Since this is a single selection field, we will be using radio buttons:

1. Browse to the View Custom Fields page.
2. Click on the Add Custom Field link.
3. Select the Radio Buttons custom field type.
4. Give the custom field the name of Is Escalation Required.
5. Accept the default options and click on Finish.
6. Select Default Screen and click on Update.

The second custom field is a simple text-based field, which will indicate what level of escalation the ticket is currently at. We do not want users (support or business) to be able to change the values as this should be determined by the system automatically, so we will be using a read-only text field. We will make use of this field in later chapters.

1. Browse to the View Custom Fields page.
2. Click on the Add Custom Field link.
3. Select the Read-only Text Field custom field type.
4. Give the custom field the name of Escalation Level.

For More Information:
5. Accept the default options and click on **Finish**.
6. Select **Default Screen** and click on **Update**.

Finally, the third custom field will contain a list of users from JIRA's user base who will receive notifications when the ticket is being escalated:

1. Browse to the **View Custom Fields** page.
2. Click on the **Add Custom Field** link.
3. Select **Multi User Picker** custom field type.
4. Give the custom field the name **Escalation List**.
5. Accept the default options and click on **Finish**.
6. Select **Default Screen** and click on **Update**.

Now that we have created the necessary custom fields, the next step is to configure them. Remember our **Is Escalation Required** custom field will allow users to specify if the tickets they raised need escalation, so we need to add the options of **Yes** and **No** to the field. We also need to set the default all tickets to require escalation:

1. Browse to the **View Custom Fields** page.
2. Click on the **Configure** link for the **Is Escalation Required** custom field.
3. Click on **Edit Options**.
4. Add the options of **Yes** and **No**, click on the **Done** button when finished.
5. Click on the **Edit Default Value** link.
6. Select the **Yes** option for the default value.

### Setting up the field configuration

Now that we have our custom fields, the next step is to create a new field configuration so that we can specify the behaviors of our custom fields. Previously, we had decided to make the **Is Escalation Required** field required, so there will be no ambiguity when it comes to determining if a ticket needs to be escalated. So, let’s start with creating a new field configuration first.

1. Browse to the **View Field Configurations** page.
2. Name the new field configuration **Help Desk Field Configuration**.
3. Provide a helpful description **Field configuration** for the help desk team.
4. Click on the **Add** button to create a new field configuration.
Now that we have our new field configuration, we can start adding configurations to our new custom fields:

1. Click on the **Configure** link for **Help Desk Field Configuration**.
2. Click on the **Required** link for the **Is Escalation Required** custom field.
   (If you do not see the **Is Escalation Required** field in the list of fields, please go back to the **View Custom Fields** page to verify whether the field has been created successfully.)

### Setting up a field configuration scheme
We have our custom fields, we have configured the relevant options, created a new field configuration, and set the behavior of our fields. It is time to add them to a scheme:

1. Browse to the **View Field Configuration Schemes** page.
2. Name the new field configuration scheme **Help Desk Field Configuration Scheme**, as we will be applying this to our Help Desk project.
3. Provide a helpful description Field configuration scheme for the help desk team.
4. Click on the **Add** button to create a new field configuration.

With the field configuration scheme in place, we can now activate our configurations. Since this is designed for our help desk team, we would want to apply the field configurations to the issue types that are applicable to the Help Desk project; that is, **Ticket** and **Incident**:

1. Click on the **Configure** link for **Help Desk Field Configuration Scheme**.
2. Select the issue type as **Ticket** and field configuration as **Help Desk Field Configuration**.
3. Click on the **Add** button to add the association.
4. Repeat steps 1 to 3 for the **Incident** issue type.
Putting it together

OK, we have done all the hard work. We have created new custom fields, a new field configuration, and a new field configuration scheme; the last step is to put everything together and see it in action:

1. Browse to the Project Administration page for our Help Desk project.
2. Click on the Select link for Field Configuration Scheme.
3. Select Help Desk Field Configuration Scheme and click on the Associate button.

Alright, we are all done! You can pat yourself on the back, sit back, and take a look at your hard work in action.

Create a new issue type Incident under the Help Desk project and you will see your new custom fields at the bottom of the page (you will not see Escalation Level, as it is read-only, so it does not appear on the Create/Edit screen).

For More Information:  
Go ahead and create the incident by filling the fields. On the View Issue page, you will see your new custom fields displayed along with the values you have provided:

<table>
<thead>
<tr>
<th>Details</th>
<th>People</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: Incident</td>
<td>Assignee: JIRA Administrator</td>
</tr>
<tr>
<td>Priority: Major</td>
<td>Reporter: JIRA Administrator</td>
</tr>
<tr>
<td>Affects Version/s: None</td>
<td>CC List: John Doe</td>
</tr>
<tr>
<td>Fix Version/s: None</td>
<td></td>
</tr>
<tr>
<td>Is Escalation Required: Yes</td>
<td>Votes: 0 ✓</td>
</tr>
<tr>
<td>Description:</td>
<td>Watchers: 0 🙋</td>
</tr>
</tbody>
</table>

When I try to access the internal staff portal, I am presented with a 500 error page.

Summary

In this chapter, we have looked at fields in JIRA. We have looked at how JIRA is able to extend its ability to capture user data through custom fields. We have also explored how we can specify different behavior for fields under different contexts through the use of field configurations and schemes. In the next chapter, we will expand on what we have learned about fields by formally introducing you to screens, and how to combine fields and screens to provide your users with the most natural and logical forms, to assist them in creating and logging issues.
Where to buy this book

You can buy JIRA 5.2 Essentials from the Packt Publishing website:

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.