JIRA Essentials

Third Edition

Use the features of JIRA to manage projects and effectively handle bugs and software issues

Patrick Li
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 3 'Issue Management'
- A synopsis of the book’s content
- More information on JIRA Essentials Third Edition

About the Author

Patrick Li is a cofounder and senior engineer at AppFusions. AppFusions is an expert in developing and packaging integrated solutions for many enterprise applications and platforms, including IBM Connections, Jive, Atlassian, Google Apps, Box, Dropbox, and more.

He has worked in the Atlassian ecosystem for over 8 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 advices on forums such as Atlassian Answers and Quora.

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.
JIRA Essentials  
*Third Edition*

Over the years, JIRA has grown from a simple bug tracking system designed for engineers to manage their projects to an all-purpose issue tracking solution. As it has matured over time, JIRA has become more than an application, it has transformed into a platform, with a suite of other products that are built on it, enabling it to adapt and deliver value to a wide variety of use cases.

JIRA 6 is the latest major release of JIRA and its product family. It comes with new features that include support for service desk applications and enhancements made to existing features such as providing flexible and robust workflow design and agile methodology support. In this book, we will cover all the basics of JIRA and its core capabilities as a feature-rich, issue-tracking system, as well as add-ons that add additional features to the JIRA platform.

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. The first chapter starts with setting up your own JIRA, and the subsequent chapters will introduce key features and concepts. With each chapter, you will learn important concepts such as business processes, workflows, e-mails, and notifications. You will also 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 the starting point of the book and aims to guide you to set up a local copy of a 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 changes in JIRA 6. By the end of this 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 is also introduced as it is the core concept of JIRA administration.

*Chapter 3, Issue Management*, covers everything related to issue creation and the operations that can be performed on an issue (excluding workflow transitions). Furthermore, this chapter touches on the various aspects of issues, as they are the focal point of JIRA. This chapter also serves 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. This chapter then continues 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 ties in 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 cycles is introduced, and various aspects of workflows explained. This chapter also explores the relationship between workflows and other various JIRA aspects that have been previously covered, such as screens. The concept of JIRA extensions is also briefly touched 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 featured in this book. This chapter also ties in the workflow chapter and explains 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 are touched on, explaining how security can be applied to each. It also covers LDAP integration, where you can hook up your JIRA with an existing LDAP system for user management.

Chapter 9, Searching, Reporting, and Analysis, focuses on how data captured in JIRA can be retrieved to provide various types of reporting features. It also covers the changes introduced in JIRA 6.

Chapter 10, JIRA Service Desk, covers the new JIRA Service Desk product, an add-on to transform JIRA into a fully-fl edged service desk solution. It looks at installing add-ons, setting up service desks, and defining custom SLA metrics.

Chapter 11, Advanced Features, covers advanced features that can help to change your JIRA into more than just a traditional issue tracking system. It looks at how you can run agile projects with JIRA through the use of JIRA Agile and how you can turn JIRA into an effective feedback collection system.
In the previous chapter, you saw that JIRA is a very flexible and versatile tool that can be used in different organizations for different purposes. A software development organization will use JIRA to manage its software development lifecycle and bug tracking, while a customer services organization may choose to use JIRA to track and log customer complaints and suggestions. For these reasons, issues in JIRA can represent anything that is applicable to the real-world scenario. Generally speaking, an issue in JIRA often represents a unit of work that can be acted upon by one or more people.

In this chapter, we will explore the basic and advanced features offered by JIRA for you to manage issues. By the end of this chapter, you will have learned the following:

- Issues and what they are in JIRA
- Creating, editing, and deleting issues
- Moving issues between projects
- Expressing your interest in issues through voting and watching
- Advanced issue operations, including uploading attachments and linking issues
Understanding issues
In JIRA, an issue can represent any number of things. In fact, an issue in a given project may mean something that is very different in another project. So what an issue really is depends on the context of what project it is in, and how you choose to define and use JIRA. For example, an issue in a normal software development project would often represent a software bug, while in a help desk project it can represent a support request.

Despite all the different objects an issue can represent, there are a number of key aspects that are common for all issues in JIRA, as follows:

- An issue must belong to a project.
- It must have a type, otherwise known as an issue type, which indicates what the issue is representing.
- It must have a summary. The summary acts like a one-line description of what the issue is about.
- It must have a status. A status indicates where along the workflow the issue is at a given time. We will discuss workflows in Chapter 6, Workflows and Business Processes.

JIRA issue summary
As we have discussed, an issue in JIRA can be anything in the real world to represent a unit of work or a task to be completed. It can be a software bug, a help desk ticket, or a customer request. However, what does an issue look like in JIRA? How does JIRA achieve this level of flexibility and still present it in a consistent manner?

Let's first take a look at an issue in JIRA. The following screenshot shows a typical example of an issue and breaks it down into more digestible sections, followed by an explanation of each of the highlighted sections in a table. This view is often called the Issue Summary or the View Issue page.
Chapter 3

Welcome to your first issue
An issue is an item you want to track in a specific JIRA project. You can use JIRA to track all kinds of things.

Types of issues
An issue could represent a software bug, a user story, a project task, a helpdesk ticket, or a leave request form; the list is endless.

Issue attributes
You may have noticed that an issue has some system fields such as type, priority, and status. These fields are helpful to help you categorize and track the progress of each issue.

Next: Changing an issue's status

Learn more about issues

Attachments

attachments

13 KB 10 Aug 14 2:36 PM

Issue Links

relates to

DEMO-4 Editing issues

Resolved

Sub-Tasks

1. Keyboard shortcuts

sub-tasks

Unassigned

Activity

All Comments Work Log History Activity Source Reviews

Patrick Li added a comment - 07 Sep 14 9:17 PM
Please attach a screenshot of the error message to the issue

[ 65 ]
These sections are described in the following table:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project / Issue Key</td>
<td>It shows the project the issue belongs to. The issue key is the unique identifier of the current issue. This section acts as a breadcrumb for easy navigation.</td>
</tr>
<tr>
<td>Issue Summary</td>
<td>It is a brief summary of the issue.</td>
</tr>
<tr>
<td>Issue View Options</td>
<td>These are the various view options for the issue. The options include XML, Excel, and Word.</td>
</tr>
<tr>
<td>Issue Operations</td>
<td>These are the operations that users can perform on the issue, such as edit, assign, and comment. These are covered in the later sections of this chapter.</td>
</tr>
<tr>
<td>Workflow Options</td>
<td>These are the workflow transitions available. Workflow will be covered in Chapter 6, Workflows and Business Processes.</td>
</tr>
<tr>
<td>Issue Details / Fields</td>
<td>This section lists the issue fields such as issue type and priority. Custom fields are also displayed in this section. Fields will be covered in Chapter 4, Field Management.</td>
</tr>
<tr>
<td>User Fields</td>
<td>This section is specific for user-type fields such as assignee and reporter. Fields will be covered in Chapter 4, Field Management.</td>
</tr>
<tr>
<td>Date Fields</td>
<td>This section is specific for date-type fields such as create and due date. Fields will be covered in Chapter 4, Field Management.</td>
</tr>
<tr>
<td>Vote / Watch Issue</td>
<td>These are the options that allow users to vote and watch an issue.</td>
</tr>
<tr>
<td>Attachments</td>
<td>These list all the attachments in an issue.</td>
</tr>
<tr>
<td>Comments</td>
<td>These list all the comments that are visible to the current user.</td>
</tr>
</tbody>
</table>

Working with issues

As we have seen, issues are the center of JIRA. In the following sections, we will look at what you, as a user, can do with issues. Note that each of the actions will require you to have specific permissions, which we will cover in Chapter 8, Securing JIRA.

Creating an issue

When creating a new issue, you will need to fill in a number of fields. Some fields are mandatory, such as the issue's summary and type, while others are optional, such as the issue's description. We will discuss fields in more details in the next chapter.

There are several ways in which you can create a new issue in JIRA. You can choose any of the following options:

- Click on the Create button at the top of the screen.
Press C on your keyboard.

This will bring up the Create Issue dialog box, as shown in the preceding screenshot. As you can see, there are quite a few fields, and the required fields will have a red asterisk (*) mark next to their names.

The administrator configures what fields will be part of the create dialog, but as a user, you can customize and make your own create screen by hiding the optional fields, by performing the following steps:

1. Click on the Configure Fields option at the top-right corner.
2. Select the Custom option.
3. Uncheck all the fields you want to hide, and check the fields that you want to show.

You are only hiding or showing these fields for yourself. Only the JIRA administrator can actually hide and show fields globally for all users.

There is a Create another option beside the Create button. By ticking this option and then clicking on the Create button, the Create Issue dialog box will stay on the screen and remember the values you have previously entered, such as priority, components, and due dates. This way, you can avoid having to fill in the whole dialog box again and will only have to update some of the fields that actually are different, such as Summary. With this feature, you can rapidly create many issues in a much shorter time frame.

**Editing an issue**

There are two ways in which you can edit an issue in JIRA. The first and more traditional way is by clicking the **Edit** button or pressing **E** on your keyboard. This will bring up the Edit Issue dialog with all the editable fields for the current issue. This allows you to make changes to multiple fields at once.
The second option is called **in-line editing**. With this feature, you will be able to view the issue and edit the field you want on the spot, without having to wait for the edit dialog to load. Scroll down to find the field. To edit a field in-line, all you have to do is hover your mouse over the value for the field you want to update, wait for the **Edit** icon to show up, click the icon, and start editing.

---

### Deleting an issue

You can delete issues from JIRA. You might need to delete issues that have been created by mistake or if the issue is redundant, although normally, it is better to close and mark the issue as a duplicate. We will discuss closing an issue in *Chapter 6, Workflows and Business Processes*.

Issue deletion is permanent in JIRA. Unlike some other applications that may put deleted records in a trash bin, which you can retrieve later, JIRA completely deletes the issue from the system. The only way to retrieve the deleted issue is by restoring JIRA with a previous backup.

Perform the following steps to delete an issue:

1. Browse to the issue you wish to delete.
2. Click on the Delete option from the More menu. This will bring up the Delete Issue dialog box.
3. Click on the Delete button to remove the issue permanently from JIRA.

Deleting an issue permanently removes it from JIRA, along with all of its data including attachments and comments.

Moving an issue between projects
Once an issue has been created, the issue is associated with a project. You can, however, move the issue around from one project to another. This may sound like a very simple process, but there are many steps involved and things to be considered.

First, you need to decide on a new issue type for the issue, if the current issue type does not exist in the new project. Second, you will need to map a status of the issue. Third, you will need to decide on the values for the fields that exist in the new project but which do not exist in the current project if those fields are set to mandatory in the new project. Sounds like a lot? Luckily, JIRA comes with a wizard that is designed to help you address all those items.

Perform the following steps to start moving an issue:
1. Browse to the issue you wish to move.
2. Click on the Move option in the More menu. This will bring up the Move Issue wizard.

There are essentially four steps in the Move Issue wizard.

The first step is to select which project you wish to move the issue to. You will also need to select the new issue type. If the same issue type exists in the new project, you can usually continue to use the same issue type.

Move Issue
The second step allows you to map the current issue to the new project's workflow. If the issue's status exists in the target project, the wizard will skip this step.

The third step shows all the fields that exist in the new project but not the current project and which require a value. Again, if there are no missing fields, this step will be skipped.
The fourth and last step shows you the summary of the changes that will be applied, by moving the issue from project A to project B. This is your last chance to make sure that all the information is correct. If there are any mistakes, you can go back to step one and start over again. If you are happy with the changes, confirm the move by clicking on Move.

Once the issue is moved, it will be given a new issue key based on the new project. However, JIRA is still able to redirect you if you access the issue with its old issue key.

Casting a vote on an issue

The most straightforward way to express your interest in a JIRA issue is to vote for it. For organizations or teams that manage their priorities based on popularity, voting is a great mechanism to collect this information. The Popular Issues project tab mentioned in Chapter 2, Project Management will display all the voted issues.

An example of this is how Atlassian uses JIRA (for example, https://jira.atlassian.com/browse/JRA-9) as a way to let its customers choose and vote for the features they want to be implemented or bugs to be fixed, by voting on issues based on their needs. This allows the product management and marketing team to have an insight on the market needs and how to best evolve their offerings.

One thing to keep in mind is when voting, you can only vote ONCE per issue. You can vote many times for many different issues, but for any given issue, you have only one vote. This helps prevent a single user from continuously voting on the same issue, which may blow the final statistics out of proportion. You can, however, unvote a vote you have already cast on an issue, and vote for it again later; if you choose to do this, it will still only count as one vote.
To vote for an issue, simply click on the Vote for this issue link next to Votes. When you have voted for an issue, the icon will appear as colored. When you have not yet voted for an issue, the icon will appear gray. Note that you cannot vote for issues you have created.

Receiving notifications on an issue

JIRA is able to send automated e-mail notifications about updates on issues to users. Normally, notification e-mails will only be sent out to the issue's reporter, assignee, and people who have registered interest in the issue. This behavior can be changed through Notification Schemes, which we will discuss in Chapter 7, E-mails and Notifications.

You can register your interest in the issue by choosing to watch the issue. By watching an issue, you will receive e-mail notifications on activity updates. Users watching the issue can also choose to stop watching, thus stop receiving e-mail updates from JIRA. You can also add other users as watchers by adding them to the watcher's list.

To watch an issue, simply click on the Start watching this issue link. When you are already watching the issue, the text will be Stop watching this issue. If you click on the link again, you will stop watching the issue, as shown in the following screenshot:

![Image of JIRA interface showing watch issues feature]

JIRA will automatically add you as a watcher for issues created by you.
JIRA also shows how many people are actively watching the issue by displaying the total watchers next to the watch icon. You can click on the number next to **Watchers** to see the full list of watchers.

### Assigning issues to users

Once an issue has been created, the user normally assigned to the issue will start working on it. Afterwards, the user can assign the issue further, for example, to QA staff for further verification.

There are many instances where an issue needs to be reassigned to a different user. For example, the current assignee may be unavailable, or if issues are created with no specific assignees. Another example will be that issues are assigned to different people at different stages of the workflow. For this reason, JIRA allows users to reassign issues once they have been created.

Perform the following steps to assign an issue:

1. Browse to the issue you wish to assign.
2. Click on the **Assign** button in the **Issue** menu bar or press A on your keyboard (you can also use the in-line edit feature here). This will bring up the **Assign** dialog.
3. Select the new assignee for the issue, and optionally add a comment to provide some information to the new assignee.
4. Click on the **Assign** button.

<table>
<thead>
<tr>
<th>Assign</th>
<th>Patrick Li</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment</td>
<td>Can you please look into this as soon as possible?</td>
</tr>
</tbody>
</table>

Shortcut tip: Pressing a also opens this dialog box.
Once this issue has been reassigned, its assignee value will be updated to the new user. The new assignee will also receive a notification e-mail, alerting him/her of the assignment. You can also un-assign an issue this way by simply selecting the **Unassigned** option. Unassigned issues do not have an assignee and will not show up on anyone's list of active issues.

You can press \( I \) on your keyboard to quickly assign the issue to yourself.

### Sharing issues with other users

If you want to e-mail an issue to other users in JIRA, instead of having to manually copy and paste the issue's URL in an e-mail, you can use the built-in share feature in JIRA. All you have to do is go to the issue you want to share, click on the share icon or press \( S \) on your keyboard as shown in the following screenshot. Then select the users you want to share the issue with and click the **Share** button.

If the user you are sharing the issue with does not have access to the issue, he/she will not be able to see the issue's details.
Issue linking

JIRA allows you to create custom hyper-links for issues. This allows you to provide more information to the issue. For example, for a support-type issue, you can link it to a web page that contains information on how to resolve similar types of problems. Another example will be to link the issue to other related issues.

Enabling issue linking

Issue linking is enabled by default when you first install JIRA. Issue linking is configured globally, so once it is enabled, it will become available for all projects in JIRA.

If issue linking is disabled, you can enable it by the following steps to enable issue linking. Note that you need to be a JIRA administrator:

1. Log in as a JIRA administrator user.
2. Browse to the JIRA administration console.
3. Select the System tab and then the Issue Linking option. This will take you to the Issue Link Administration page.
4. Click on Activate to enable issue linking in JIRA.

From this page, you will be able to see a list of Link Types available. A link type defines the nature of the link between issues. As shown in the screenshot in the next section, the default Blocks link type defines that issue A blocks issue B. This means that issue B cannot be completed until issue A is resolved. It is important to note that issue linking does not place any restrictions on issues. So in our example, although the link states that issue B cannot be completed until issue A is done, this is not enforced and users can close issue A while issue B remains open.

All the link types shown here are for standard issue links. There are no remote issue link types. However, you do need to enable issue linking to make use of remote issue links.

Creating link types

JIRA comes with four link types by default: Blocks, Clones, Duplicate, and Relates. As we have discussed earlier, a link type is simply a type of association or a label on the association, which describes the relationships between two issues. Therefore, you can define your own link types in JIRA. For example, one issue can support another issue, and we can create a link type called Supports.
Perform the following steps to create a new link type:

1. Browse to the *Issue Linking Administration* page.
2. Make sure issue linking is enabled.
3. Fill in the **Add New Link Type** form. Do the following for the **Support** link type:
   1. Enter *Supports* for the **Name** field.
   2. Enter *supports* for the **Outward Link Description** field.
   3. Enter *is supported by* for the **Inward Link Description** field.
4. Click on the **Add** button.

Outward and inward link descriptions define what is shown when users select the type of link to use when linking two issues together. We will see an example of this in a later section. Once a new link type is added, it will be displayed in the table on the *Issue Linking Administration* page.
Linking issues with other issues

Issues are often related to other issues in some way. For example, issue A might be blocking issue B, or issue C might be a duplicate of issue D. You can add descriptions to the issue to capture this information, or delete one of the issues in the duplication case, but with this approach, it is hard to keep a track of all these relationships. Luckily, JIRA provides an elegant solution for this, with the standard issue link feature.

The **standard issue link** lets you link an issue with one or more other issues in the same JIRA instance. So, you can link two issues from different projects together (if you have access to both the projects). Linking issues in this way is very simple; all you need to know is the target issues to link to:

1. Browse to the **View Issue** page for the issue you wish to create a link for.
2. Select **Link** from the **More Actions** menu. This will bring up the **Link Issue** dialog box.
3. Select the **JIRA Issue** option from the left panel.
4. Select the type of issue linking.
5. Select the issues to link to. You can use the search facility to help you locate the issues you want.
6. Click on the **Link** button.
After you have linked your issues, they will be displayed in the Issue Links section on the View Issue page. JIRA will display the target issue's key, description, priority, and status.

### Linking issues with remote contents

The standard JIRA issue link allows you to link multiple issues in the same JIRA instance. JIRA also lets you link issues to resources such as a web page on the Internet.

Using remote issue links is very similar to the standard issue link; the difference is that instead of selecting another issue, the URL address of the target resource is specified:

1. Open up the Link Issue dialog box.
2. Select the Web Link option from the left panel.
3. Specify the URL address for the target resource. JIRA will automatically try to find and load the appropriate icon for the resource.
4. Provide the name for the link in Link Text field. The name you provide here will be what is shown for the link when viewing the issue.
5. Click on the Link button.

<table>
<thead>
<tr>
<th>JIRA Issue</th>
<th>Web Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter a URL to link this issue to</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://en.wikipedia.org/wiki/JIRA">http://en.wikipedia.org/wiki/JIRA</a></td>
</tr>
<tr>
<td>Link Text</td>
<td>Wikipedia Article</td>
</tr>
<tr>
<td>Comment</td>
<td>JIRA on Wikipedia</td>
</tr>
</tbody>
</table>

[79]
Issue cloning

When you need to create a new issue and you already have a baseline issue, JIRA allows you to quickly create it with the data based on your existing issues, by cloning the original one. Cloning an issue allows you to quickly create a new one with most of its fields populated. For example, you might have two software products with the same bug. After creating a bug report in one project, you can simply clone it for the other project.

A cloned issue will have all the fields copied from the original issue; however, it is a separate entity nonetheless. Further actions performed on either of the two issues will not affect the other.

When an issue is being cloned, a Clone link is automatically created between the two issues, establishing a relationship.

Cloning an issue in JIRA is simple and straightforward. All you have to do is specify a new summary (or accept the default summary with CLONE—text at the front) for the cloned issue:

1. Browse to the issue you wish to clone.
2. Select Clone from the More Actions menu. This will bring up the Clone Issue page.
3. Type in a new summary for the new cloned issue.
4. Check the Clone Attachments checkbox if you also want to copy over all the attachments.
5. Click on the Create button.

Once the issue is successfully cloned, you will be taken to the issue summary page for the newly cloned issue.
Time tracking

Since issues often represent a single unit of work that can be worked on, it is logical for users to log the time they have spent working on it. You can specify an estimated effort required to complete an issue, and JIRA will be able to help you track the progress. JIRA displays the time tracking information of an issue in the **Time Tracking** panel at the right-hand side, as shown in the following screenshot with its description:

<table>
<thead>
<tr>
<th>Time Tracking</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated:</td>
<td>![1w]</td>
</tr>
<tr>
<td>Remaining:</td>
<td>![3d]</td>
</tr>
<tr>
<td>Logged:</td>
<td>![2d]</td>
</tr>
</tbody>
</table>

- **Estimated**: This represents the original estimated effort required to complete the issue. For example, the estimated time required for fixing a bug by a developer.
- **Remaining**: This represents the remaining time for the issue to be completed. It is calculated automatically by JIRA based on the original estimate and total time logged by users. However, the user logging work on the issue, as described in the following section, can also override this value.
- **Logged**: This represents the total time spent on the issue so far.

Configuring time tracking

Time tracking is enabled by default when you install JIRA. To change your time tracking settings, you will first need to deactivate it, and then activate it again with the new settings:

1. Log in as a JIRA administrator user.
2. Browse to the JIRA administration console.
3. Select the **System** tab, and then select the **Time Tracking** option. This will take you to the **Time Tracking** page.
4. Click on the **Deactivate** button.
5. Configure the time tracking parameters.
6. Click on **Activate** to enable time tracking in JIRA.

![Time Tracking is currently OFF.](image)

Activate Time Tracking below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours per day</td>
<td>This is the number of working hours per day in JIRA. For example, if this is set to 8 hours per day, when the user puts in 16 hours, JIRA will automatically convert it to 2 days.</td>
</tr>
<tr>
<td>Days per week</td>
<td>This is the number of working days per week in JIRA. For example, if this is set to 5 days per week, when the user puts in 10 days, JIRA will automatically convert it to 2 weeks.</td>
</tr>
<tr>
<td>Time format</td>
<td>This is the format in which JIRA will display time in the Time Spent field.</td>
</tr>
<tr>
<td>Default Unit</td>
<td>This is the default time tracking unit if the user does not supply one.</td>
</tr>
<tr>
<td>Legacy Mode</td>
<td>This will make JIRA use the old time tracking behavior.</td>
</tr>
<tr>
<td>Copy Comment To Work Description</td>
<td>This is a very handy feature, where JIRA will automatically copy comments made during a workflow transition (workflow will be discussed in Chapter 6, Workflows and Business Processes) that allows time logging.</td>
</tr>
</tbody>
</table>

You can set several parameters on the **Time Tracking** page. As we will see in later sections, these parameters will determine the time tracking behavior for JIRA, as shown in the following table:
Specifying original estimates

**Original estimate** represents the anticipated time required to complete the work represented by the issue. It is shown as the blue bar under the Time Tracking section.

In order for you to specify an original estimate value, you need to make sure that time tracking is enabled and the Time Tracking field is added to the issue's create and/or edit screen. We will discuss fields and screens in Chapter 4, Field Management and Chapter 5, Screen Management, respectively.

To specify an original estimate value, provide a value for the **Original Estimate** field on the create issue and/or edit issue screen.

Logging work

Logging work in JIRA allows you to specify the amount of time (work) you have spent working on an issue. You can log work against any of the issues, provided that Time Tracking is enabled and you have the permission to do so. We will cover permissions in Chapter 9, Searching, Reporting, and Analysis.

Perform the following steps to log work against an issue:

1. Browse to the issue you wish to log work against.
2. Select Log Work from the More Actions menu. This will bring up the Log Work page.
3. Enter the amount of time you wish to log. Use w, d, h, and m to specify week, day, hour, and minute, respectively.
4. Select the date you wish to log your work against.
5. Optionally, select how the remaining estimate should be adjusted.
6. Add a description to the work you have done.
7. Optionally, select who can view the work log entry.
8. Click on the **Log** button.

When you log work on an issue, you have the option to choose how the **Remaining Estimate** value will be affected. By default, this value will be automatically calculated by subtracting the amount logged from the original estimate. You can, however, choose other options available, such as setting the remaining estimate to a specific value or reducing it by an amount that is different from the amount of work being logged.

You can also click on the + sign in the **Time Tracking** section to log time.
Issues and comments

JIRA lets users create comments on issues. As we have already seen, you will be able to create comments when assigning an issue to a different user. This is a very useful feature that allows multiple users to collaborate to work on the same issue and share information. For example, the support staff (issue assignee) may request more clarification from the business user (issue reporter) by adding a comment to the issue. When combined with JIRA’s built-in notification system, automatic e-mail notifications will be sent to the issue’s reporter, assignee, and any of the other users watching the issue. Notifications will be covered in Chapter 7, E-mails and Notifications.

Adding comments

By default, all logged-in users will be able to add comments to issues they can access. Perform the following steps to add a comment to an issue:

1. Browse to the issue you wish to add a comment to.
2. Click on the Comment option in the Issue menu bar or press M on your keyboard. This will bring up the Comment input section.
3. Type a comment into the text box. The text box will adjust its size as you type.
4. Click on the Add button to add the comment.
Once a comment has been added, the comment will be visible in the Comments tab in the Activity section. When you are creating comments, you can select who can view your comment using the comment access control. This is very useful if you have external users viewing the issue and you only want to share your comments with internal users.

Managing your comments
After you have added your comment to an issue, you can edit its contents and security settings or delete it altogether. To edit or delete a comment, simply hover over the comment, and the comment management option will appear to the right-hand side.

Permalinking a comment
From time to time, you will want to refer other people to a comment you made previously. While you can tell them the issue and let them scroll down to the bottom until they find your comment amongst hundreds of others, JIRA allows you to create a quick permalink to your comment that will take you directly to the comment of interest.

Perform the following steps to create a permalink for a comment:

1. Browse to the comment you wish to permalink.
2. Hover over the comment to bring up the comment management options.
3. Click on the permalink icon. This will highlight the comment in pale blue.
You will now notice that your browser's URL bar will look something similar to http://sample.jira.com/browse/DEMO-1?focusedCommentId=10100&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-10100 as a sample link (notice the focusedCommentId section after the issue key). Copy and paste that URL and give it to your colleagues; once they click on this link, they will be taken directly to the highlighted comment.

**Attachments**

As we have seen so far, JIRA uses fields such as summary and description to capture data. This works for most cases, but when you have complex data such as application log files or screenshots, fields become insufficient. This is where attachments come in. JIRA allows you to attach files to issues as support documents.

**Enabling attachments in JIRA**

Attachments are saved as files on the JIRA file server and not in the database, so you need to ensure that there is sufficient disk space to accommodate the volume of attachments for now and future growth. As attachments are not stored in the database, JIRA will not backup the files as a part of its backup process. Attachments need to be backed up separately.

Attachments are enabled by default in JIRA, so users will be able to attach files to issues as soon as JIRA is installed. However, if it is disabled for some reason, you can re-enable it. Perform the following steps to enable attachments for JIRA:

1. Log in to JIRA as a JIRA administrator.
2. Browse to the JIRA administration console.
3. Select the System tab and then the Attachments option. This will take you to the Attachment Settings page.
4. Click on the Edit Settings button. This will bring up the Edit Attachment Settings page.
5. Select Use Default Directory for the Attachment Path option.
6. Click on the Update button to enable attachments in JIRA.
On the **Attachment Settings** page, as shown in the following screenshot, there are a few options that you will need to configure while enabling attachments in JIRA:

![Edit Attachment Settings](image)

The following table summarizes the configuration options:

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attachment Path</strong></td>
<td>This specifies the location where attachments will be stored on the filesystem. The only options are the default directory, which is inside your JIRA_HOME directory, or disabled attachments in JIRA.</td>
</tr>
<tr>
<td><strong>Attachment Size</strong></td>
<td>This specifies the maximum size of the attachment that users can upload. Default is 10 MB.</td>
</tr>
<tr>
<td><strong>Enable Thumbnails</strong></td>
<td>This specifies whether or not to enable thumbnail generation when the attachment is an image.</td>
</tr>
<tr>
<td><strong>Enable ZIP support</strong></td>
<td>This specifies whether or not to enable ZIP support, which allows users to download multiple attachments as a single ZIP file, and also view the contents of the ZIP attachment files.</td>
</tr>
</tbody>
</table>
Attachments are enabled and disabled globally across JIRA. You cannot selectively enable or disable attachment functions on a per project basis. You can, however, achieve a similar result by controlling the permission around who can attach files. Permissions are discussed in Chapter 8, Securing JIRA.

**Attaching files**

JIRA allows you to attach any arbitrary file to issues. These can be image files, Microsoft Office documents, and other binary files. Perform the following steps to attach a file to an issue:

1. Browse to the issue you wish to attach a file.
2. Select **Attach File** from the **More** menu. This will bring up the **Attach Files** dialog.
3. Click on the **Browse** button to select the file you wish to attach. You can repeat this step to attach more than one file.
4. Optionally, provide a comment for the attached file. The comment will be added as a normal comment to the issue.
5. Click on the **Attach** button.
Attaching screenshots

Apart from letting you attach any file to an issue, JIRA also allows you to directly attach a screenshot from your system clipboard to issues. This saves you from having to take a screenshot, save it as a physical file on the disk, and finally attach it to JIRA.

Perform the following steps to attach a screenshot:

1. Take a screenshot with your operating system. For example, if you are on Windows, press the Print Screen key.
2. Browse to the issue you wish to attach a screenshot.
3. Select Attach screenshot from the More menu. This will bring up the Attach screenshot dialog.
4. Press the Ctrl + V keys on your keyboard, and the screenshot will be pasted into the panel above.
5. Enter a filename for the screenshot or accept the default name.
6. Click on the Upload button.

Issue types and subtasks

As seen earlier, issues in JIRA can represent many things ranging from software development tasks to project management milestones. Issue type is what differentiates one kind of issue from another.

Each issue has a type (therefore, the name issue type), which is represented by the issue type field. This lets you know what type of issue it is, and also helps you determine many other aspects of it, such as what fields will be displayed for this issue.

JIRA comes with a set of default issue types, as shown in the following table:

<table>
<thead>
<tr>
<th>Issue type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug</td>
<td>A problem that impairs or prevents the functions of the product</td>
</tr>
<tr>
<td>Improvement</td>
<td>An enhancement to an existing feature</td>
</tr>
<tr>
<td>New feature</td>
<td>A new feature of the product</td>
</tr>
<tr>
<td>Task</td>
<td>A task that needs to be done</td>
</tr>
</tbody>
</table>
The default issue types are great for simple software development projects, but they do not necessarily meet the needs of others. Since it is impossible to create a system that can address everyone's needs, JIRA lets you create your own issue types and assigns them to projects. For example, for a help desk project, you might want to create a custom issue type called ticket. You can create this custom issue type and assign it to the Help Desk project, and users will be able to log tickets, instead of bugs in the system.

Issue types are managed through the **Manage Issue Types** page. Perform the following steps to access this page:

1. Log in to JIRA as a JIRA administrator.
2. Browse to the JIRA administration console.
3. Select the Issues tab and then the Issue Types option. This will take you to the Issue Types Administration page.

The preceding screenshot shows a list of default issue types along with custom issue types created by the administrator.
Creating issue types
You can create any number of issue types. Perform the following steps to create a new issue type:

1. Browse to the Manage Issue Types page.
2. Click on the Add Issue Type button.
3. Type a unique name for the new issue type.
4. Type in a general description for the issue type.
5. Select whether the new issue type will be a standard issue type or a subtask issue type.
6. Click on Add to create the new issue type.

Once the new issue type is created, it will be assigned a default icon. If you want to change the icon, you will need to click on the Edit link for the issue type, and then select a new image as its icon.

Deleting issue types
When deleting an issue type, you have to keep in mind that the issue type might already be in use; meaning there are issues created with that issue type. So, if you delete an issue type, you will need to select a new one for those issues. The good news is that JIRA takes care of this for you. As shown in the following screenshot, we delete the Bug issue type and JIRA informs us of the already existing 19 issues of type Bug. You will need to assign them to a new issue type, such as Improvement.

<table>
<thead>
<tr>
<th>Delete Issue Type: Bug</th>
<th>click to view the 19 issues using this issue type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note: This issue type cannot be deleted - there are currently 19 matching issues with no suitable alternative issue types (only issues you have permission to see will be displayed, which may be different from the total count shown on this page). In order for an issue type to be deleted, it needs to be associated with one workflow, field configuration and field screen scheme across all projects. If this is not the case, JIRA can not provide a list of valid replacement issue types.</td>
<td></td>
</tr>
<tr>
<td>Cancel</td>
<td></td>
</tr>
</tbody>
</table>
Perform the following steps to delete an existing issue type:

1. Browse to the Manage Issue Types page.
2. Click on the Delete link for the issue type you wish to delete. This will bring up the Delete Issue Type page. If there are existing issues with the issue type you are trying to delete, you will be asked to select a new issue type.
3. Select the new issue type for the existing issues. Once deleted, those issues will be automatically updated to the new issue type.
4. Click on the Delete button to delete the issue type.

Subtasks
JIRA allows only one person (assignee) to work on one issue at a time. This design ensures that an issue is a single unit of work that can be tracked against one person. However, in the real world, we often find ourselves in situations where we need to have multiple people working on the same issue. This may be caused by a poor breakdown of tasks or simply because of the nature of the task at hand. Whatever the reason, JIRA provides a mechanism to address this problem through subtasks.

Subtasks are similar to issues in many ways, and as a matter of fact, they are a special kind of issue. They must have a parent issue, and their issue types are flagged as subtask issue types. You can say that all subtasks are issues, but not all issues are subtasks.

For every issue, you can have one or more subtasks that can be assigned and tracked separately from another. Subtasks cannot have other subtasks. JIRA allows only one level of subtasks.

Enabling subtasks
Subtasks are enabled by default. If you have subtasks disabled for some reason, you will need to enable this feature.

Perform the following steps to enable subtasks in JIRA:

1. Browse to the JIRA administration console.
2. Select the Issues tab and then the Sub-Tasks option. This will take you to the Sub-Tasks Administration page.
3. Click on Enable to enable subtasks in JIRA.
Creating subtasks
Since subtasks belong to an issue, you need to browse to the issue first before you can create a new subtask:

1. Browse to the issue you wish to create subtasks for.
2. Select Create Sub-Task from the More Actions menu.

You will see a familiar Create Issue dialog box. However, one thing you will notice is that, unlike when you are creating an issue, you do not select which project to create the subtask in. This is because JIRA can determine the project value based on the parent issue. You will also notice that you can only select issue types that are subtasks.

Other than these differences, creating a subtask is no different than creating a normal issue. You can customize the fields shown in the dialog box and choose to rapidly create multiple subtasks by selecting the Create another option.

Once the subtask has been created, it will be added to the Sub-Tasks section of the parent issue. You will see all the subtasks that belong to the issue and their status. If a subtask has been completed, it will have a green tick next to it.

<table>
<thead>
<tr>
<th>Sub-Tasks</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ✅ Verifying hard disk is not failing</td>
<td>DONE Patrick Li</td>
</tr>
<tr>
<td>2. ✅ Check BIOS settings</td>
<td>DONE Patrick Li</td>
</tr>
<tr>
<td>3. Re-install operating system</td>
<td>IN PROGRESS Patrick Li</td>
</tr>
</tbody>
</table>

Issue type schemes
Issue type schemes are templates or collections of issue types that can be applied to projects. As shown in the following screenshot, JIRA comes with Default Issue Type Scheme, which is applied to all projects that do not have specific issue type schemes applied.
When a new issue type is created in JIRA, it is added to **Default Issue Type Scheme**. This means that the new issue type will be available to all the projects by default. This will become a problem when you start to have specialized projects such as help desk, and issue types such as **Bug** become inappropriate.

To overcome this problem, JIRA lets us group a collection of issue types together. We can also rearrange the order of the issue types within the collection so that they appear in the drop-down list in a logical manner to the users. In effect, we create an issue type scheme, a template for issue types, and their order, which can be reused and applied to one or more projects.

We will create a new issue type scheme for our example of JIRA implementation later in this chapter.

### Creating issue type schemes

Perform the following steps to create a new issue type scheme:

1. Browse to the administration console.
2. Select the **Issues** tab and then the **Issue Type Schemes** option. This will bring you to the **Issue Type Schemes** page.
3. Click on the **Add Issue Type Scheme** button.
4. Provide a name for the new issue type scheme in **Scheme Name**.
5. Drag the issue types you want to be part of the scheme from the **Available Issue Types** list and drop them into the **Issue Types for Current Scheme** list.
6. Select a **Default Issue Type** value. Note that this is optional, and you can only select a default issue type after you have selected at least one issue type for the new scheme.

7. Click on the **Save** button.

After the issue type scheme has been created, you will need to associate it with the projects you want to apply the scheme to. The simplest way to do this is to click on the **Associate** link for the issue type scheme, and then select the projects you want. This allows you to apply the scheme to multiple projects at once. If there are already issues of issue type that does not belong to the new issue type scheme, JIRA will prompt you and walk you through to change the issue type value for all the affected issues.


**Issue priorities**

Priorities help users to set the importance of issues. Users can first assign priority values to issues and later use it to sort the list of issues they have to work on. Thus, helping the team decide which issues to focus on first. JIRA comes with five levels of priorities out of the box, as shown in the following screenshot:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Icon</th>
<th>Color</th>
<th>Order</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocker</td>
<td>Blocks development and/or testing work, production could not run.</td>
<td>☝️</td>
<td>🔴</td>
<td>⬇️</td>
<td>Edit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical</td>
<td>Crashes, loss of data, severe memory leak.</td>
<td>⬆️</td>
<td>🔴</td>
<td>⬆️</td>
<td>Edit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Major</td>
<td>Major loss of function.</td>
<td>⬆️</td>
<td>🔴</td>
<td>⬆️</td>
<td>Edit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minor</td>
<td>Minor loss of function, or other problem where easy workaround is present.</td>
<td>⬇️</td>
<td>🔴</td>
<td>⬆️</td>
<td>Edit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trivial</td>
<td>Cosmetic problem like misspelt words or misaligned text.</td>
<td>⬇️</td>
<td>🔴</td>
<td>⬆️</td>
<td>Edit</td>
</tr>
</tbody>
</table>

You can customize this list by creating your own priorities. To create new priorities, follow these steps:

1. Browse to the administration console.
2. Select the **Issues** tab and then the **Priorities** option.
3. Specify the name of the new priority.
4. Click the **select image** link to choose an icon for the priority.
5. Specify a color for the priority. You can either type in HTML color hex code directly or use the color picker to help you select the color you want. The color chosen here will be used when icon images cannot be displayed, such as when you export issues to a spreadsheet.

6. Click on the Add button.

Priorities are global. This means that all projects will share the same set of priorities.

The help desk project

In this exercise, we will continue our setup for the project we have created in the previous chapter. We will add the following configurations to our project:

- A set of new issue types that are specific to our help desk project
- A new scheme to limit the selection of issue types

Adding new issue types

Since our project is for a help desk support team, the default issue types that come with JIRA are not appropriate for this purpose. For this reason, let's create our own issue types and associate them with the project. For this exercise, we will create two new issue types, incident and ticket.

The first step to set up an issue type association is to create the two issue types we need, incident and ticket:

1. Browse to the Issue Types page.
2. Click on the Add Issue Type button.
3. Type in Incident for the Name field.
4. Click on Add to create the new Incident issue type.

You should now see the new Incident issue type in the table. Now, let's add the Ticket issue type:

1. Type in Ticket for the Name field.
2. Click on Add to create the new Ticket issue type.
You should see both the Incident and Ticket issue types. However, this will only make our new issue types available, but will not make them the only options when creating a new issue for our project. Default issue types, such as Bug and New Feature, which are not applicable for a help desk, are still available. By leaving them there, we are running the risk of confusing the users and allowing mistakes to be made.

If you remember from previous discussions, we can address this problem with a new issue type scheme. Let’s go ahead and create one.

Creating an issue type scheme

We want to limit the issue types to be only Incident and Ticket for our Global Help Desk project, but we do not want to affect the other projects that still need to have Bug and other default issue types. So, we need to create a new issue type scheme specifically for support projects, which can be used by us and other teams:

1. Browse to the Issue Type Schemes page.
2. Click on the Add Issue Type Scheme button.
3. Name our new issue type scheme Support Desk Issue Type Scheme.
4. Drag the Incident and Ticket issue types from the Available Issue Types panel to the Issue Types for Current Scheme panel.
5. Select Incident as the Default Issue Type value.
6. Click on the Save button.

With the issue type scheme in place, the last step is to apply it to our Support Desk project:

1. Click on the Associate link for the Help Desk Issue Type Scheme option.
2. Select your Global Help Desk project.
3. Click on the Associate button to apply this issue type scheme to the project.

If you have issues in the Global Helpdesk project with issue types that do not exist in our new issue type scheme, follow the issue type change wizard and change those issues to either Incident or Ticket.
Putting it together

With everything created and set up, you can go back and create a new issue to see how it all looks. If everything works out, you should see something similar to the following screenshot:

As you can see, Incident and Ticket are the only issue types that can be selected when creating a new issue for the Global Help Desk project.

Summary

In this chapter, we looked at what issues are in JIRA and explored the basic operations of creating, editing, and deleting issues. We also looked at the advanced operations offered by JIRA to enhance how you can manipulate and use issues, such as adding attachments, creating subtasks, linking multiple issues, and so on.

In the next chapter, we will look at fields and what we can do with them.
Where to buy this book

You can buy JIRA Essentials Third Edition from the Packt Publishing website.

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

Click here for ordering and shipping details.